IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION, VOL. 46, NO. 11, NOVEMBER 1998

Integral Equations and Discretizations for Waveguide Apertures

John J. Ottusch, George C. Valley, and Stephen Wandzura

Abstract—**We present integral equations and their discretizations for calculating the fields radiated from arbitrarily shaped antennas fed by cylindrical waveguides of arbitrary cross sections. We give results for scalar fields in two dimensions with Dirichlet and Neumann boundary conditions and for (vector) electric and magnetic fields in three dimensions. The discretized forms of the equations are cast in identical format for all four cases. Feed modes can be TM, TE, or transverse electromagnetic (TEM). A method for numerically computing the modes of an arbitrarily shaped, cylindrical waveguide aperture is also given.**

Index Terms—**Aperture antennas, integral equations.**

I. INTRODUCTION

NUMERICAL simulation of the electromagnetic performance of antennas using integral equations requires a mathematical model of the driving sources. In contrast to scattering cross-section computations where a distant source creates a plane wave in the vicinity of the scatterer, construction of an accurate source model for an antenna is nontrivial. If a simple approach, such as a "delta-gap" excitation [1] is used, the accuracy of some important antenna parameters, such as input impedance, gain, and reflection can be seriously compromised, even for cases in which the far-field pattern is obtained accurately.

The purpose of this paper is twofold. First, we develop integral equations representing exact specification of the field emanating from an aperture of arbitrary shape with the field entering the aperture left unconstrained and to be determined. The exact definition of the "emanating" field is accomplished by analysis of a translationally invariant waveguide that has the cross section of the given aperture. In the context of a generalized scattering problem such as a waveguide-fed antenna, such an integral equation may serve as a boundary condition that must be obeyed inside the waveguide on any plane normal to its axis. Second, we derive discretized forms of the integral equations[1] (using the method of moments) that are suitable for numerical computation. As part of this development, we give a useful interpretation of the kernel that appears in the "waveguide integral equation."

Our development is based on the assumption that the waveguide is:

- translationally invariant in the half-space behind the aperture along the axis normal to the aperture;
- terminated by a perfect absorber or is so long as to be practically nonreflecting;
- filled with a linear, isotropic, homogeneous medium;
- enclosed by walls that are infinitely hard or infinitely soft in the scalar scattering case or perfectly conducting in the electromagnetic scattering case.

The first section is devoted to finding continuous and discretized forms of the waveguide integral equations for scalar waves and then applying them to more general scattering problems. These equations apply to acoustic scattering in two or three dimensions as well as the two-dimensional (2-D) analogues of three-dimensional (3-D) electromagnetic scattering (which apply to scatterers with translational symmetry in a direction orthogonal to the axis of the waveguide). In the second section, we do the same for 3-D electromagnetic scattering. The two treatments are entirely analogous. Formulas for the power flow out of (due to the given excitation) and into (due to back scattering) the waveguide are also given in each section. In the third section, we show how the waveguide integral equations can be extended to more general circumstances. Prescriptions for numerically computing the modes of cylindrical waveguides with arbitrary cross sections may be found in the Appendix.

II. SCALAR WAVEGUIDE EQUATIONS

A. Modes

An arbitrary field $\psi(\mathbf{x})$ that satisfies the scalar Helmholtz equation

$$(\nabla^2 + k^2)\psi(\mathbf{x}) = 0 \tag{1}$$

inside a waveguide aligned with the $\mathbf{z}$ axis, can be written as a sum of modal components[2] traveling in the $+\hat{\mathbf{z}}$ and $-\hat{\mathbf{z}}$ directions [3]

$$\psi(\mathbf{x}_\perp, z) = \sum_n (a_n e^{i\beta_n z} + b_n e^{-i\beta_n z}) u_n(\mathbf{x}_\perp). \tag{2}$$

Manuscript received September 22, 1997; revised July 24, 1998.
J. J. Ottusch and S. M. Wandzura are with the Communications and Photonics Laboratory, HRL Laboratories, Malibu, CA 90265 USA.
G. C. Valley is with the Hughes Space and Communications Company, Los Angeles, CA 90009 USA.
Publisher Item Identifier S 0018-926X(98)08896-6.

[1] An equivalent formulation of the feed model for the electromagnetic case has been used previously by McGrath and Pyati [2] We, however, try to clarify the intent, development, and use of this formulation in the context of a generalized method of moments discretization.

[2] For simplicity, we will assume that no cutoff modes (i.e., those with $\beta = 0$) are present. It is straightforward to amend the development to handle such modes.

0018–926X/98$10.00 © 1998 IEEE

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

DTIC QUALITY INSPECTED 4

20000817 068

Likewise, the longitudinal derivative of the field may be written as

$$\frac{\partial\psi(\mathbf{x}_\perp, z)}{\partial z} = \sum_n (a_n e^{i\beta_n z} - b_n e^{-i\beta_n z})\frac{ik}{Z_n}u_n(\mathbf{x}_\perp) \qquad (3)$$

where

$$Z_n = \frac{k}{\beta_n} \qquad (4)$$

is the modal impedance. In these equations, an implicit $e^{-i\omega t}$ time dependence is assumed for the fields. $k = \omega/c$ is the free-space propagation constant and β_n and $u_n(\mathbf{x}_\perp)$ are, respectively, the propagation constant and transverse field distribution of the nth mode inside the guide. The modes are eigensolutions to the scalar wave equation

$$(\nabla_\perp^2 + k^2 - \beta_n^2)u_n(\mathbf{x}_\perp) = 0 \qquad (5)$$

for $\mathbf{x}_\perp$ inside the waveguide aperture W and the $u_n(\mathbf{x}_\perp)$ are constrained to satisfy the boundary conditions of the waveguide walls when $\mathbf{x}_\perp$ is on the boundary of the aperture ∂W. With proper normalization, the modes form a complete and orthonormal set of functions over W, i.e.,

$$\sum_n u_n(\mathbf{x}_\perp)u_n(\mathbf{x}'_\perp) = \delta(\mathbf{x}_\perp - \mathbf{x}'_\perp) \quad \text{Completeness} \qquad (6)$$

and

$$\int_W d\mathbf{x}_\perp u_m(\mathbf{x}_\perp)u_n(\mathbf{x}_\perp) = \delta_{mn} \quad \text{Orthonormality.} \qquad (7)$$

B. Waveguide Integral Equation

Let $\psi^{\text{out}}(\mathbf{x}_\perp, z)$ denote a specified outgoing wave, $z = 0$ correspond to the plane of the waveguide aperture, and the rest of the waveguide be located in the half-space with $z < 0$. Using the modal expansions and the completeness relation for the modes, we can write the following expression for $\psi^{\text{out}}(\mathbf{x}_\perp, 0)$ in terms of the field and its longitudinal derivative on W:

$$\begin{aligned}\psi^{\text{out}}(\mathbf{x}_\perp, 0) &= \sum_n a_n u_n(\mathbf{x}_\perp)\\ &= \frac{1}{2}\sum_n (a_n + b_n)u_n(\mathbf{x}_\perp)\\ &\quad + \frac{1}{2}\sum_n \frac{Z_n}{ik}(a_n - b_n)\frac{ik}{Z_n}u_n(\mathbf{x}_\perp)\\ &= \frac{1}{2}\psi(\mathbf{x}_\perp, 0) + \frac{1}{2}\int_W d\mathbf{x}'_\perp H(\mathbf{x}_\perp, \mathbf{x}'_\perp)\\ &\quad \times \left.\frac{\partial\psi(\mathbf{x}'_\perp, z')}{\partial z'}\right|_{z'=0} \end{aligned} \qquad (8)$$

where

$$H(\mathbf{x}_\perp, \mathbf{x}'_\perp) = \sum_n \frac{Z_n}{ik}u_n(\mathbf{x}_\perp)u_n(\mathbf{x}'_\perp). \qquad (9)$$

For any point $\mathbf{x}$ on a general surface S, we may define an independent surface field quantity

$$\sigma(\mathbf{x}) \equiv -\lim_{\mathbf{x}'\to\mathbf{x}} \hat{\mathbf{n}}(\mathbf{x})\cdot\nabla'\psi(\mathbf{x}'); \quad \mathbf{x} \text{ on } S \qquad (10)$$

where $\hat{\mathbf{n}}(\mathbf{x})$ is the outward unit normal to S at $\mathbf{x}$. In the case of a waveguide aperture, σ simplifies to

$$\sigma(\mathbf{x}_\perp, 0) \equiv -\left.\frac{\partial\psi(\mathbf{x}'_\perp, z')}{\partial z'}\right|_{z'=0}; \quad \mathbf{x}_\perp \text{ on } W. \qquad (11)$$

Inserting this into (8) and dropping the spatial coordinate z, we obtain the following integral equation on the waveguide aperture that relates the field, its longitudinal derivative, and the specified waveguide excitation on W:

$$2\psi^{\text{out}}(\mathbf{x}_\perp) = \psi(\mathbf{x}_\perp) - \int_W d\mathbf{x}'_\perp H(\mathbf{x}_\perp, \mathbf{x}'_\perp)\sigma(\mathbf{x}'_\perp). \qquad (12)$$

$H(\mathbf{x}_\perp, \mathbf{x}'_\perp)$ is the kernel of the "square root" of the transverse wave operator in the sense that

$$\int_W d\mathbf{x}'_\perp H(\mathbf{x}_\perp, \mathbf{x}'_\perp)H(\mathbf{x}'_\perp, \mathbf{x}''_\perp) = \hat{G}_\perp(\mathbf{x}_\perp, \mathbf{x}''_\perp) \qquad (13)$$

where $\hat{G}_\perp$ obeys

$$(\nabla_\perp^2 + k^2)\hat{G}_\perp(\mathbf{x}_\perp, \mathbf{x}''_\perp) = -\delta(\mathbf{x}_\perp - \mathbf{x}''_\perp) \qquad (14)$$

inside the waveguide and satisfies the boundary conditions on the waveguide walls.

A different relation between ψ, σ, and the outgoing wave is obtained if we specify $\partial\psi^{\text{out}}(\mathbf{x}_\perp, z)/\partial z$ instead of $\psi^{\text{out}}(\mathbf{x}_\perp, z)$ to write

$$\begin{aligned}\frac{\partial\psi^{\text{out}}(\mathbf{x}_\perp, 0)}{\partial z} &= \sum_n a_n\frac{ik}{Z_n}u_n(\mathbf{x}_\perp)\\ &= \frac{1}{2}\sum_n (a_n + b_n)\frac{ik}{Z_n}u_n(\mathbf{x}_\perp)\\ &\quad + \frac{1}{2}\sum_n \frac{ik}{Z_n}(a_n - b_n)u_n(\mathbf{x}_\perp)\\ &= \frac{1}{2}\left.\frac{\partial\psi(\mathbf{x}'_\perp, z')}{\partial z'}\right|_{z'=0} + \frac{1}{2}\int_W d\mathbf{x}'_\perp \tilde{H}(\mathbf{x}_\perp, \mathbf{x}'_\perp)\\ &\quad \times \psi(\mathbf{x}_\perp, 0) \end{aligned} \qquad (15)$$

where[3]

$$\tilde{H}(\mathbf{x}_\perp, \mathbf{x}'_\perp) = \sum_n \frac{ik}{Z_n}u_n(\mathbf{x}_\perp)u_n(\mathbf{x}'_\perp). \qquad (16)$$

Dropping the spatial coordinate z and defining σ as before, we get an alternative form for the waveguide integral equation

$$2\frac{\partial\psi^{\text{out}}(\mathbf{x}_\perp)}{\partial z} = \frac{\partial\psi(\mathbf{x}_\perp)}{\partial z} + \int_W d\mathbf{x}'_\perp \tilde{H}(\mathbf{x}_\perp, \mathbf{x}'_\perp)\psi(\mathbf{x}'_\perp) \qquad (17)$$

or

$$-2\frac{\partial\psi^{\text{out}}(\mathbf{x}_\perp)}{\partial z} = \sigma(\mathbf{x}_\perp) - \int_W d\mathbf{x}'_\perp \tilde{H}(\mathbf{x}_\perp, \mathbf{x}'_\perp)\psi(\mathbf{x}'_\perp). \qquad (18)$$

$\tilde{H}(\mathbf{x}_\perp, \mathbf{x}'_\perp)$ and $H(\mathbf{x}_\perp, \mathbf{x}'_\perp)$ are "inverse operators" in the sense that

$$\int_W d\mathbf{x}'_\perp H(\mathbf{x}_\perp, \mathbf{x}'_\perp)\tilde{H}(\mathbf{x}'_\perp, \mathbf{x}''_\perp) = \delta(\mathbf{x}_\perp - \mathbf{x}''_\perp). \qquad (19)$$

---

[3] Note that $\tilde{H}(\mathbf{x}_\perp, \mathbf{x}'_\perp)$ is not a *function* since the sum over all n does not converge. Rather, like the Dirac delta "function" $\delta(\mathbf{x}_\perp, \mathbf{x}'_\perp)$, it is a *distribution*, which, when convolved with a suitably smooth function, produces a well-defined value.

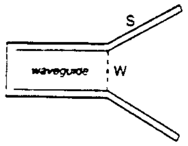


Fig. 1. Antenna system composed of waveguide aperture W and antenna surface S.

Using (12) and (18) on the waveguide aperture W, we can derive boundary integral equations that apply to more general scattering cases. For example, we can write coupled boundary integral equations for the case of a waveguide aperture connected to a general scatterer. This is demonstrated in the next subsection for the special cases in which the scattering surface obeys either Dirichlet or Neumann boundary conditions. In both cases, it is assumed that the union of the scatterer S and waveguide aperture W forms a closed surface, as indicated in Fig. 1.

C. Coupled Integral Equations

In this section, we derive integral equations relating the known field emanating from the waveguide aperture to an unknown surface field (either ψ or σ) for the generic closed antenna system shown in Fig. 1. For Dirichlet (Neumann) boundary conditions on S, the unknown surface field on both S and W is chosen to be $\sigma(\psi)$.

1) Dirichlet Boundary Conditions on S: The integral equation for the field (in the absence of an explicit incident wave) is [4]

$$\frac{1}{2}\psi(\mathbf{x}) = \oint_{S \ominus W} ds'\{[\hat{\mathbf{n}}(\mathbf{x}') \cdot \nabla' G(\mathbf{x}, \mathbf{x}')]\psi(\mathbf{x}') - G(\mathbf{x}, \mathbf{x}')\sigma(\mathbf{x}')\} \tag{20}$$

for $\mathbf{x}$ on $S \ominus W$. The Helmholtz kernel $G(\mathbf{x}, \mathbf{x}')$ is given by

$$G(\mathbf{x}, \mathbf{x}') = \begin{cases} \frac{i}{4} H_0^{(1)}(k|\mathbf{x} - \mathbf{x}'|) & \text{in } 2d \\ \frac{e^{ik|\mathbf{x}-\mathbf{x}'|}}{|\mathbf{x}-\mathbf{x}'|} & \text{in } 3d \end{cases} \tag{21}$$

where $H_0^{(1)}$ is the zeroth-order Hankel function of the first kind. For Dirichlet boundary conditions on S (i.e., $\psi(\mathbf{x} \text{ on } S) = 0$) we have

$$0 = \oint_{S \ominus W} ds' G(\mathbf{x}, \mathbf{x}')\sigma(\mathbf{x}') + \int_W ds'[\hat{\mathbf{n}}(\mathbf{x}') \cdot \nabla' G(\mathbf{x}, \mathbf{x}')]\psi(\mathbf{x}') \tag{22}$$

for $\mathbf{x}$ on S and

$$\frac{1}{2}\psi(\mathbf{x}) = \oint_{S \ominus W} ds' G(\mathbf{x}, \mathbf{x}')\sigma(\mathbf{x}') + \int_W ds'[\hat{\mathbf{n}}(\mathbf{x}') \cdot \nabla' G(\mathbf{x}, \mathbf{x}')]\psi(\mathbf{x}') \tag{23}$$

for $\mathbf{x}$ on W. Equations (22) and (23) along with either (12) or (18) form a set of coupled integral equations to be solved for $\psi(\mathbf{x})$ on W and $\sigma(\mathbf{x})$ on $S \oplus W$. Using (12) we can eliminate ψ, putting the known field $\psi^{\text{out}}(\mathbf{x})$ on the left and the unknown quantity $\sigma(\mathbf{x})$ on the right

$$\begin{aligned} -2\oint_W ds'(\hat{\mathbf{n}}' \cdot \nabla' G(\mathbf{x}, \mathbf{x}'))\psi^{\text{out}}(\mathbf{x}') &= \oint_S ds' G(\mathbf{x}, \mathbf{x}')\sigma(\mathbf{x}') - \oint_u ds' G(\mathbf{x}, \mathbf{x}')\sigma(\mathbf{x}') \\ &\quad - \oint_W ds'(\hat{\mathbf{n}}' \cdot \nabla' G(\mathbf{x}, \mathbf{x}')) \int_W ds'' H(\mathbf{x}', \mathbf{x}'')\sigma(\mathbf{x}'') \end{aligned} \tag{24}$$

for x on S and

$$\begin{aligned} \psi^{\text{out}}(\mathbf{x}) - 2\oint_W ds'(\hat{\mathbf{n}}' \cdot \nabla' G(\mathbf{x}, \mathbf{x}'))\psi^{\text{out}}(\mathbf{x}') &= \oint_S ds' G(\mathbf{x}, \mathbf{x}')\sigma(\mathbf{x}') \\ &\quad + \oint_W ds'[G(\mathbf{x}, \mathbf{x}')\sigma(\mathbf{x}') - \frac{1}{2}H(\mathbf{x}, \mathbf{x}')\sigma(\mathbf{x}')] \\ &\quad + \oint_W ds'(\hat{\mathbf{n}}' \cdot \nabla' G(\mathbf{x}, \mathbf{x}')) \int_W ds'' H(\mathbf{x}', \mathbf{x}'')\sigma(\mathbf{x}'') \end{aligned} \tag{25}$$

for x on W.

2) Neumann Boundary Conditions on S: The integral equation for σ (i.e. the normal derivative of the field) may be written as [4]

$$\frac{1}{2}\sigma(\mathbf{x}) = -(\hat{\mathbf{n}}(\mathbf{x}) \cdot \nabla) \oint_{S \ominus W} ds'\{[\hat{\mathbf{n}}(\mathbf{x}') \cdot \nabla' G(\mathbf{x}, \mathbf{x}')]\psi(\mathbf{x}') + G(\mathbf{x}, \mathbf{x}')\sigma(\mathbf{x}')\} \tag{26}$$

or

$$\begin{aligned} \frac{1}{2}\sigma(\mathbf{x}) = \oint_{S \ominus W} ds'\{&[\hat{\mathbf{n}}(\mathbf{x}) \times \nabla G(\mathbf{x}, \mathbf{x}')] \cdot [\hat{\mathbf{n}}(\mathbf{x}') \times \nabla' \psi(\mathbf{x}')] \\ &- k^2(\hat{\mathbf{n}}(\mathbf{x}) \cdot \hat{\mathbf{n}}(\mathbf{x}'))G(\mathbf{x}, \mathbf{x}')\psi(\mathbf{x}') \\ &- \hat{\mathbf{n}}(\mathbf{x})\nabla G(\mathbf{x}, \mathbf{x}')\sigma(\mathbf{x}')\} \end{aligned} \tag{27}$$

for $\mathbf{x}$ on $S \ominus W$. The first form is more compact (and for that reason is employed below), the second more convenient for numerical computation. For Neumann boundary conditions on S (i.e., $\sigma(\mathbf{x} \text{ on } S) = 0$), we have

$$\begin{aligned} 0 = &-(\hat{\mathbf{n}}(\mathbf{x}) \cdot \nabla) \int_S ds'[\hat{\mathbf{n}}(\mathbf{x}') \cdot \nabla' G(\mathbf{x}, \mathbf{x}')]\psi(\mathbf{x}') \\ &- (\hat{\mathbf{n}}(\mathbf{x}) \cdot \nabla) \int_W ds'\{[\hat{\mathbf{n}}(\mathbf{x}') \cdot \nabla' G(\mathbf{x}, \mathbf{x}')]\psi(\mathbf{x}') \\ &+ G(\mathbf{x}, \mathbf{x}')\sigma(\mathbf{x}')\} \end{aligned} \tag{28}$$

for $\mathbf{x}$ on S and

$$\begin{aligned} \frac{1}{2}\sigma(\mathbf{x}) = &-(\hat{\mathbf{n}}(\mathbf{x}) \cdot \nabla) \int_S ds'[\hat{\mathbf{n}}(\mathbf{x}') \cdot \nabla' G(\mathbf{x}, \mathbf{x}')]\psi(\mathbf{x}') \\ &- (\hat{\mathbf{n}}(\mathbf{x}) \cdot \nabla) \int_W ds'\{[\hat{\mathbf{n}}(\mathbf{x}') \cdot \nabla' G(\mathbf{x}, \mathbf{x}')]\psi(\mathbf{x}') \\ &+ G(\mathbf{x}, \mathbf{x}')\sigma(\mathbf{x}')\} \end{aligned} \tag{29}$$

for $\mathbf{x}$ on W. Combining (28) and (29) with (18), we can eliminate σ and write the following integral equations for $\psi(\mathbf{x})$

in terms of the known quantity $\partial\psi^{\text{out}}(\mathbf{x})/\partial z$:

$$\begin{aligned}2\int_W ds'(\hat{\mathbf{n}}\cdot\nabla G(\mathbf{x},\mathbf{x}'))\frac{\partial\psi^{\text{out}}}{\partial z}(\mathbf{x}') &= (\hat{\mathbf{n}}\cdot\nabla)\int_S ds'(\hat{\mathbf{n}}'\cdot\nabla' G(\mathbf{x},\mathbf{x}'))\psi(\mathbf{x}')\\ &\quad-(\hat{\mathbf{n}}\cdot\nabla)\int_W ds'(\hat{\mathbf{n}}'\cdot\nabla' G(\mathbf{x},\mathbf{x}'))\psi(\mathbf{x}')\\ &\quad+\int_W ds'(\hat{\mathbf{n}}\cdot\nabla G(\mathbf{x},\mathbf{x}'))\int_W ds''\tilde{H}(\mathbf{x}',\mathbf{x}'')\psi(\mathbf{x}'')\end{aligned} \tag{30}$$

for x on S and

$$\begin{aligned}\frac{\partial\psi^{\text{out}}}{\partial z}(\mathbf{x})-2\int_W ds'(\hat{\mathbf{n}}\cdot\nabla G(\mathbf{x},\mathbf{x}'))\frac{\partial\psi^{\text{out}}}{\partial z}(\mathbf{x}') &= (\hat{\mathbf{n}}\cdot\nabla)\int_S ds'(\hat{\mathbf{n}}'\cdot\nabla' G(\mathbf{x},\mathbf{x}'))\psi(\mathbf{x}')\\ &\quad-(\hat{\mathbf{n}}\cdot\nabla)\int_W ds'(\hat{\mathbf{n}}'\cdot\nabla' G(\mathbf{x},\mathbf{x}'))\psi(\mathbf{x}')\\ &\quad-\frac{1}{2}\int_W ds'\tilde{H}(\mathbf{x},\mathbf{x}')\psi(\mathbf{x}')\\ &\quad-\int_W ds'(\hat{\mathbf{n}}\cdot\nabla G(\mathbf{x},\mathbf{x}'))\int_W ds''\tilde{H}(\mathbf{x}',\mathbf{x}'')\psi(\mathbf{x}'')\end{aligned} \tag{31}$$

for x on W.

D. Discretization

While analytical solutions for waveguide modes are known for a few special cross sections, in general, modes must be computed numerically. Even when analytical solutions exist, it is more convenient (from a computational perspective) to use numerical solutions because then all interacting surfaces, whether physical or intangible (e.g. waveguide apertures), can be treated equivalently.

Assume the waveguide aperture has been discretized into a set of patches that support M basis functions $f_m(\mathbf{x})$. Following the procedure given in the Appendix, we can write approximate expressions for the N lowest waveguide modes in terms of basis functions defined on the aperture

$$u_n(\mathbf{x}) = \sum_{m=1}^{M} A_{nm} f_m(\mathbf{x}). \tag{32}$$

In the usual method of moments fashion, we approximate the field ψ and its normal derivative σ on the aperture as linear combinations of the basis functions with unknowns coefficients S_m^W and I_m^W

$$\psi(\mathbf{x}) \approx \sum_{m=1}^{M} S_m^W f_m(\mathbf{x}) \tag{33}$$

$$\sigma(\mathbf{x}) \approx \sum_{m=1}^{M} I_m^W f_m(\mathbf{x}). \tag{34}$$

We also approximate $H(\mathbf{x},\mathbf{x}')$ as a truncated sum over the N computed modes

$$H(\mathbf{x},\mathbf{x}') \approx \sum_{n=1}^{N} \frac{Z_n}{ik} u_n(\mathbf{x}) u_n(\mathbf{x}'). \tag{35}$$

Then, by substituting (32)–(35) into (12), and applying the testing operator $\int_W ds f_i(\mathbf{x})\cdot$ to both sides of the resultant equation, we arrive at the discretized form of (12)

$$2V^W = N^W S^W - X^W I^W \tag{36}$$

where

$$V_i^W = \int_W ds\,\psi^{\text{out}}(\mathbf{x}) f_i(\mathbf{x}) \tag{37a}$$

$$N_{ij}^W = \int_W ds f_i(\mathbf{x}) f_j(\mathbf{x}) \tag{37b}$$

$$\begin{aligned}X_{ij}^W &= \int_W ds\int_W ds' f_i(\mathbf{x}) H(\mathbf{x},\mathbf{x}') f_j(\mathbf{x}')\\ &= [(AN^W)^T \Lambda (AN^W)]_{ij}\end{aligned} \tag{37c}$$

and

$$\Lambda_{mn} = \frac{Z_n}{ik}\delta_{mn}. \tag{38}$$

A similar procedure produces the discretized form of (18), namely

$$2\tilde{V}^W = N^W I^W - \tilde{X}^W S^W \tag{39}$$

where

$$\tilde{V}_i^W = -\int_W ds \frac{\partial\psi^{\text{out}}}{\partial z}(\mathbf{x}) f_i(\mathbf{x}) \tag{40a}$$

$$\begin{aligned}\tilde{X}_{ij}^W &= \int_W ds\int_W ds' f_i(\mathbf{x})\tilde{H}(\mathbf{x},\mathbf{x}') f_j(\mathbf{x}')\\ &= [(AN^W)^T \tilde{\Lambda}(AN^W)]_{ij}\end{aligned} \tag{40b}$$

and

$$\tilde{\Lambda}_{mn} = \frac{ik}{Z_n}\delta_{mn} = (\Lambda^{-1})_{mn}. \tag{41}$$

Equations (12) and (18) and their discretized equivalents (36) and (39) may be viewed as nonlocal inhomogeneous boundary conditions that must be obeyed on the waveguide aperture. They are nonlocal because the "surface impedance" terms X^W and $\tilde{X}^W$ relate the field at one point on the aperture to its derivative not just at the same point, but everywhere on the aperture, and vice versa. The equations are inhomogeneous if excitations V^W and $\tilde{V}^W$ are nonzero.

The discretized forms of the coupled integral equations for Dirichlet boundary conditions on S are obtained by first approximating the source on S in terms of basis functions as

$$\sigma(\mathbf{x}) \approx \sum_{m=1}^{M} I_m^S f_m(\mathbf{x}) \tag{42}$$

then substituting this approximation and the approximate expressions for $\psi(\mathbf{x})$, $\sigma(\mathbf{x})$, and $H(\mathbf{x}_\perp,\mathbf{x}'_\perp)$ on W into (22) and (23) and finally applying the testing function operator $\int_{S\cup W} ds f_i(\mathbf{x})\cdot$ to both sides. The result in block matrix form is

$$\begin{bmatrix}-2Y^{SW}(N^W)^{-1}V^W\\ V^W\end{bmatrix} = \begin{bmatrix}Z^{SS} & Z^{SW}+Y^{SW}(N^W)^{-1}X^W\\ Z^{WS} & Z^{WW}-\frac{1}{2}X^W\end{bmatrix}\begin{bmatrix}I^S\\ I^W\end{bmatrix} \tag{43}$$

where

$$Y^{SW}_{ij} = \int_S ds \int_W ds' f_i(\mathbf{x})(\hat{\mathbf{n}}' \cdot \nabla' G(\mathbf{x}, \mathbf{x}')) f_j(\mathbf{x}') \tag{44}$$

$$Z^{\alpha\beta}_{ij} = \int_\alpha ds \int_\beta ds' f_i(\mathbf{x}) G(\mathbf{x}, \mathbf{x}') f_j(\mathbf{x}') \tag{45}$$

with S or W replacing α and β.

An analogous result is obtained for the case of Neumann boundary conditions on S. We approximate the source on S as

$$\psi(\mathbf{x}) \approx \sum_{m=1}^{M} S^S_m f_m(\mathbf{x}) \tag{46}$$

substitute this expression and the approximate expressions for $\psi(\mathbf{x})$, $\sigma(\mathbf{x})$, and $\tilde{H}(\mathbf{x}_\perp, \mathbf{x}'_\perp)$ on W into (28) and (29) and then apply the testing operator. The result is

$$\begin{bmatrix} 2Y^{SW}(N^W)^{-1}V^W \\ -\tilde{V}^W \end{bmatrix} = \begin{bmatrix} \tilde{Z}^{SS} & \tilde{Z}^{SW} - \tilde{Y}^{SW}(N^W)^{-1}\tilde{X}^W \\ \tilde{Z}^{WS} & \tilde{Z}^{WW} + \frac{1}{2}\tilde{X}^W \end{bmatrix} \begin{bmatrix} S^S \\ S^W \end{bmatrix} \tag{47}$$

where

$$\tilde{Y}^{SW}_{ij} = \int_S ds \int_W ds' f_i(\mathbf{x})(\hat{\mathbf{n}} \cdot \nabla G(\mathbf{x}, \mathbf{x}')) f_j(\mathbf{x}') \tag{48}$$

$$\begin{aligned} \tilde{Z}^{\alpha\beta}_{ij} = \int_\alpha ds \int_\beta ds' [f_i(\mathbf{x})[\hat{\mathbf{n}}(\mathbf{x}) \times \nabla G(\mathbf{x}, \mathbf{x}')] \cdot [\hat{\mathbf{n}}(\mathbf{x}') \\ \times \nabla' f_j(\mathbf{x}')] - k^2 (\hat{\mathbf{n}}(\mathbf{x}) \cdot \hat{\mathbf{n}}(\mathbf{x}')) f_i(\mathbf{x}) G(\mathbf{x}, \mathbf{x}') f_j(\mathbf{x}')] \end{aligned} \tag{49}$$

with S or W replacing α and β.

E. Modal Decomposition

In preparation for computing the power flowing across the waveguide aperture in either direction, it is useful to write ψ and $\partial\psi/\partial z$ in terms of modes propagating in either direction.

By employing the completeness relation for the modes we can decompose the field on W into a sum over modes as

$$\psi(\mathbf{x}) = \sum_n \eta_n u_n(\mathbf{x}) \tag{50}$$

where

$$\eta_n = \int_W ds u_n(\mathbf{x}) \psi(\mathbf{x}) \tag{51}$$

is the amplitude of the nth mode contained in $\psi(\mathbf{x})$. It is useful to further decompose $\psi(\mathbf{x})$ into its incoming and outgoing components

$$\psi(\mathbf{x}) = \psi^{\text{in}}(\mathbf{x}) + \psi^{\text{out}}(\mathbf{x}). \tag{52}$$

Since the discretized representation of $\psi^{\text{out}}(\mathbf{x})$ is given by V^W, we may write the discretized form of η^{out}_n as

$$\eta^{\text{out}}_n = \sum_m A_{nm} V^W_m. \tag{53}$$

Using (12) to eliminate $\psi(\mathbf{x})$, we arrive at the discretized form of η^{in}_n

$$\eta^{\text{in}}_n = \sum_m A_{nm}(V^W + X^W I^W)_m. \tag{54}$$

Similarly, we may decompose the longitudinal derivative of the field as

$$\frac{\partial\psi(\mathbf{x})}{\partial z} = \sum_n \tilde{\eta}_n u_n(\mathbf{x}) \tag{55}$$

where

$$\tilde{\eta}_n = \int_W ds u_n(\mathbf{x}) \frac{\partial\psi(\mathbf{x})}{\partial z}. \tag{56}$$

Then, using

$$\frac{\partial\psi(\mathbf{x})}{\partial z} = \frac{\partial\psi^{\text{in}}(\mathbf{x})}{\partial z} + \frac{\partial\psi^{\text{out}}(\mathbf{x})}{\partial z} \tag{57}$$

and (18), we can write $\tilde{\eta}^{\text{out}}_n$ and $\tilde{\eta}^{\text{in}}_n$ in discretized form as

$$\tilde{\eta}^{\text{out}}_n = -\sum_m A_{nm} \tilde{V}^W_m \tag{58}$$

and

$$\eta^{\text{in}}_n = -\sum_m A_{nm}(\tilde{V}^W + \tilde{X}^W S^W)_m. \tag{59}$$

F. Power

The time-averaged power-flow density vector (the scalar equivalent to the Poynting vector) is [5]

$$\langle \mathbf{S}(\mathbf{x}) \rangle = \frac{1}{2} \operatorname{Re}[ic\omega\psi(\mathbf{x}) \nabla \psi(\mathbf{x})^*] \tag{60}$$

where c is a constant.

The total power flowing across the waveguide aperture in the $\hat{\mathbf{z}}$ direction is made up of an incoming part associated with the incoming parts of ψ and $\partial\psi/\partial z$ and an outgoing part associated with the outgoing parts of ψ and $\partial\psi/\partial z$. The total power exiting (entering) the waveguide aperture is given by

$$\begin{aligned} P^\alpha &= \int_W ds \langle \mathbf{S}^\alpha(\mathbf{x}) \cdot \hat{\mathbf{z}} \rangle \\ &= \frac{1}{2} \int_W ds \operatorname{Re}\left[ic\omega\psi^\alpha(\mathbf{x}) \frac{\partial\psi^\alpha(\mathbf{x})^*}{\partial z}\right] \end{aligned} \tag{61}$$

for $\alpha =$ out (in). This integral is most conveniently evaluated by decomposing ψ^α and $\partial\psi^\alpha/\partial z$ into their modal components. The reason is that since the modes are orthogonal, the power in the sum over modes is equal to the sum of the powers in each mode.

The amplitude of the nth outgoing (incoming) mode contained in $\psi(\mathbf{x})$ is η^{out}_n (η^{in}_n). Therefore, the time-averaged power exiting (entering) the waveguide aperture is

$$P^\alpha = c\omega k \sum_n^{n_{\max}} \frac{|\eta^\alpha_n|^2}{2Z_n} \tag{62}$$

for $\alpha =$ out (in), where $n_{\max}$ is the largest value of n for which β_n is real. We exclude modes with imaginary propagation constants since such modes do not transport any power into or out of the guide on average.

The amplitude of the nth outgoing (incoming) mode contained in $\partial\psi/\partial z$ is $\tilde{\eta}_n^{\text{out}}$ ($\tilde{\eta}_n^{\text{in}}$). Therefore, the time-averaged power exiting (entering) the waveguide aperture is

$$P^{\alpha} = c\frac{\omega}{k}\sum_{n}^{n_{\max}} \frac{Z_n|\tilde{\eta}_n^{\alpha}|^2}{2} \tag{63}$$

for α = out (in).

1) Acoustic Waves: If ψ is the velocity potential, i.e., $\mathbf{v} = \nabla\psi$, and ρ is the mass density, then the constant c in (60) is given by

$$c = \rho \tag{64}$$

Furthermore, the acoustic impedance [5] is related to our modal impedance by

$$Z_n^{\text{acoustic}} = \frac{\omega}{k}\rho Z_n. \tag{65}$$

2) Electromagnetic Waves in Two Dimensions: Suppose a waveguide whose axis is parallel to $\hat{\mathbf{z}}$ is also translationally invariant in the $\hat{\mathbf{y}}$ direction, i.e., the waveguide consists of a pair of half-infinite plates parallel to the yz plane. When a geometry is translationally invariant in one direction, the electromagnetic scattering problem can be decoupled into two independent problems, each of which is isomorphic to a 2-D scalar scattering problem with a different boundary condition. If the 3-D surfaces are perfectly conducting, the boundary conditions for the corresponding scalar fields on the corresponding 2-D surfaces become either Dirichlet or Neumann.

Solutions to the scalar waveguide problem with Dirichlet boundary conditions inside the waveguide correspond to solutions to the electromagnetic waveguide problem with exclusively TE modes inside the waveguide according to

$$\mathbf{E}(\mathbf{x}) = \psi(x)\hat{\mathbf{x}}, \quad \mathbf{H}(x) = \frac{\sigma(x)}{i\omega\mu}\hat{\mathbf{x}} \times \hat{\mathbf{z}} \quad \text{Dirichlet/TE} \tag{66}$$

and solutions to the scalar waveguide problem with Neumann boundary conditions inside the waveguide correspond to solutions to the electromagnetic waveguide problem with exclusively TM modes inside the waveguide according to

$$\mathbf{H}(\mathbf{x}) = \psi(x)\hat{\mathbf{x}}, \quad \mathbf{E}(x) = \frac{\sigma(x)}{i\omega\epsilon}\hat{\mathbf{z}} \times \hat{\mathbf{x}} \quad \text{Neumann/TM}. \tag{67}$$

Note how the correspondence between TM or TE polarization and Dirichlet or Neumann boundary conditions in the waveguide mode case differs from the correspondence between TM or TE polarization and Dirichlet or Neumann boundary conditions in the case of scattering from perfect conductors. On a perfect conductor we associate TM-polarized electromagnetic scattering with solutions to the scalar scattering problem with Dirichlet boundary conditions according to

$$\mathbf{E}(\mathbf{x}) = \psi(x)\hat{\mathbf{y}}, \quad \mathbf{H}(x) = \frac{\sigma(x)}{i\omega\mu}\hat{\mathbf{y}} \times \hat{\mathbf{n}} \quad \text{Dirichlet/TM} \tag{68}$$

and we associate TE-polarized electromagnetic scattering with solutions to the scalar scattering problem with Neumann boundary conditions according to

$$\mathbf{H}(\mathbf{x}) = \psi(x)\hat{\mathbf{y}}, \quad \mathbf{E}(x) = \frac{\sigma(x)}{i\omega\epsilon}\hat{\mathbf{n}} \times \hat{\mathbf{y}} \quad \text{Neumann/TE} \tag{69}$$

where $\hat{\mathbf{y}}$ is the direction of translational invariance and $\hat{\mathbf{n}}$ is the outward surface normal. Therefore, the waveguide-excited electromagnetic scattering problem with TM (TE) polarization in which all the scattering surfaces are perfect conductors, is equivalent to the waveguide-excited scalar problem, in which Neumann (Dirichlet) boundary conditions hold on the inner walls of the waveguide and Dirichlet (Neumann) boundary conditions hold on all the surfaces of all the scatterers.

For electromagnetic waves in two dimensions, the constant c in (60) is given by

$$c = \begin{cases} \frac{1}{\mu\omega^2} & \text{Dirichlet/TE} \\ \frac{1}{\epsilon\omega^2} & \text{Neumann/TM} \end{cases} \tag{70}$$

where μ and ϵ are appropriate to the material inside the guide.

III. ELECTROMAGNETIC WAVEGUIDE EQUATIONS

A. Modes

The electric and magnetic fields inside a waveguide with perfectly conducting walls can be decomposed into modal components just as the field and its normal derivative were in the scalar case. The essential difference is that now there are three distinct categories of modal fields, namely TM, TE, and transverse electromagnetic (TEM); each is a vector function rather than scalar function. For our purposes, it is sufficient to consider only the transverse components of the electric and magnetic fields. Assuming the guide is uniformly filled with a nondissipative medium having dielectric constant ϵ and magnetic permeability μ, we may write[4] [6]

$$\mathbf{E}_\perp(\mathbf{x}_\perp, z) = \sum_n (a_n e^{i\beta_n z} + b_n e^{-i\beta_n z})\mathbf{u}_n(\mathbf{x}_\perp) \tag{71}$$

$$\mathbf{H}_\perp(\mathbf{x}_\perp, z) = \sum_n (a_n e^{i\beta_n z} - b_n e^{-i\beta_n z})\frac{1}{Z_n}\hat{\mathbf{z}} \times \mathbf{u}_n(\mathbf{x}_\perp) \tag{72}$$

where the modal impedance Z_n is given by

$$Z_n = \sqrt{\frac{\mu}{\epsilon}} \times \begin{cases} \frac{\beta_n}{k}, & \text{for } n \in \text{TM modes} \\ 1, & \text{for } n \in \text{TEM modes} \\ \frac{k}{\beta_n}, & \text{for } n \in \text{TE modes.} \end{cases} \tag{73}$$

The modes are the eigensolutions to the transverse Helmholtz equation

$$(\nabla_\perp^2 + k^2 - \beta_n^2)\mathbf{u}_n(\mathbf{x}_\perp) = 0 \tag{74}$$

for $\mathbf{x}_\perp$ inside the waveguide aperture W and $\mathbf{u}_n(\mathbf{x}_\perp)$ constrained by the perfect electrical conductor boundary condition on ∂W. With proper normalization, the modes form a complete

---

[4] As in the scalar case, cutoff modes are neglected.

[5] $\overline{\delta}(\mathbf{x} - \mathbf{x}')$ is a tensor distribution, which, for any vector-valued surface functions $\mathbf{f}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ on W obeys

$$\int_W ds'\,\mathbf{f}(\mathbf{x})\cdot\overline{\delta}(\mathbf{x} - \mathbf{x}')\cdot\mathbf{g}(\mathbf{x}') = \mathbf{f}(\mathbf{x})\cdot\mathbf{g}(\mathbf{x}).$$

and orthonormal set of functions over W, i.e.,

$$\sum_n \mathbf{u}_n(\mathbf{x}_\perp)\mathbf{u}_n(\mathbf{x}'_\perp) = \overleftrightarrow{\delta}(\mathbf{x}_\perp - \mathbf{x}'_\perp)$$

$$\sum_n (\hat{\mathbf{z}} \times \mathbf{u}_n(\mathbf{x}_\perp))(\hat{\mathbf{z}} \times \mathbf{u}_n(\mathbf{x}'_\perp)) = \overleftrightarrow{\delta}(\mathbf{x}_\perp - \mathbf{x}'_\perp)$$

$$\text{Completeness}^5 \qquad (75)$$

and

$$\int_W d\mathbf{x}_\perp \mathbf{u}_m(\mathbf{x}_\perp) \cdot \mathbf{u}_n(\mathbf{x}_\perp) = \delta_{mn} \quad \text{Orthonormality.} \qquad (76)$$

B. Computation of Vector Modes from Scalar Functions

The TM and TE modes can be deduced from the solutions to the scalar Helmholtz equation on W with Dirichlet and Neumann boundary conditions, respectively, on ∂W [6]. The TM mode corresponding to the nth scalar waveguide mode $\varphi_n(\mathbf{x}_\perp)$ obeying Dirichlet boundary conditions on ∂W is

$$\mathbf{u}_n(\mathbf{x}_\perp) = \frac{\nabla_\perp \varphi_n(\mathbf{x}_\perp)}{\sqrt{k^2 - \beta_n^2}} \qquad (77)$$

and the TE mode corresponding to the nth scalar waveguide mode $\psi_n(\mathbf{x}_\perp)$ obeying Neumann boundary conditions on ∂W is

$$\mathbf{u}_n(\mathbf{x}_\perp) = \frac{\hat{\mathbf{z}} \times \nabla_\perp \psi_n(\mathbf{x}_\perp)}{\sqrt{k^2 - \beta_n^2}}. \qquad (78)$$

TEM modes are possible if and only if W is multiply connected, in which case they are related to solutions to the electrostatic potential problem on W. The TEM mode corresponding to the solution $\zeta_n(\mathbf{x})$ to the electrostatic potential problem on W with all except the nth boundary at zero potential is given by

$$\mathbf{u}_n(\mathbf{x}_\perp) \propto \nabla_\perp \zeta_n(\mathbf{x}_\perp). \qquad (79)$$

The scale factor should be chosen to enforce orthonormality for the TEM modes. This amounts to assigning a particular value to the otherwise arbitrary potential on the nth boundary. For all TEM modes, $\beta_n = k$.

C. Waveguide Integral Equation

Let $\mathbf{E}_\perp^{\text{out}}(\mathbf{x}_\perp, z)$ be the transverse component of electric field for a specified outgoing wave. Using the modal expansions and the first completeness relation for the modes, we can write the following expression for $\mathbf{E}_\perp^{\text{out}}(\mathbf{x}_\perp, 0)$ in terms of the transverse components of the electric and magnetic fields on W:

$$\begin{aligned}\mathbf{E}_\perp^{\text{out}}(\mathbf{x}_\perp, 0) &= \sum_n a_n \mathbf{u}_n(\mathbf{x}_\perp)\\ &= \frac{1}{2}\sum_n (a_n + b_n)\mathbf{u}_n(\mathbf{x}_\perp)\\ &\quad + \frac{1}{2}\sum_n Z_n(a_n - b_n)\frac{1}{Z_n}\mathbf{u}_n(\mathbf{x}_\perp)\\ &= \frac{1}{2}\mathbf{E}_\perp(\mathbf{x}_\perp, 0) - \frac{1}{2}\int_W d\mathbf{x}'_\perp\, \overleftrightarrow{\mathbf{H}}(\mathbf{x}_\perp, \mathbf{x}'_\perp)\\ &\quad \cdot (\hat{\mathbf{z}} \times \mathbf{H}_\perp(\mathbf{x}'_\perp, 0))\end{aligned} \qquad (80)$$

where the dyad

$$\overleftrightarrow{\mathbf{H}}(\mathbf{x}_\perp, \mathbf{x}'_\perp) = \sum_n Z_n \mathbf{u}_n(\mathbf{x}_\perp)\mathbf{u}_n(\mathbf{x}'_\perp) \qquad (81)$$

is the analogue of the scalar function $H(\mathbf{x}_\perp, \mathbf{x}'_\perp)$. Dropping the spatial coordinate z, we get the following expression for the waveguide integral equation on W, which relates the transverse components of the electric field, the magnetic field, and the specified electric field waveguide excitation on W:

$$\begin{aligned}2\mathbf{E}_\perp^{\text{out}}(\mathbf{x}_\perp) = \mathbf{E}_\perp(\mathbf{x}_\perp) - \int_W d\mathbf{x}'_\perp\, \overleftrightarrow{\mathbf{H}}(\mathbf{x}_\perp, \mathbf{x}'_\perp)\\ \cdot (\hat{\mathbf{z}} \times \mathbf{H}_\perp(\mathbf{x}'_\perp)).\end{aligned} \qquad (82)$$

Defining equivalent electric and magnetic currents on W by

$$\mathbf{J}(\mathbf{x}_\perp) = \hat{\mathbf{z}} \times \mathbf{H}_\perp(\mathbf{x}_\perp) \qquad (83)$$

$$\mathbf{M}(\mathbf{x}_\perp) = -\hat{\mathbf{z}} \times \mathbf{E}_\perp(\mathbf{x}_\perp) \qquad (84)$$

allows us to write the waveguide integral equation in terms of equivalent currents as

$$2\mathbf{E}_\perp^{\text{out}}(\mathbf{x}_\perp) = \hat{\mathbf{z}} \times \mathbf{M}(\mathbf{x}_\perp) - \int_W d\mathbf{x}'_\perp\, \overleftrightarrow{\mathbf{H}}(\mathbf{x}_\perp, \mathbf{x}'_\perp) \cdot \mathbf{J}(\mathbf{x}'_\perp). \qquad (85)$$

If $\mathbf{H}_\perp^{\text{out}}(\mathbf{x}_\perp, z)$ is specified instead of $\mathbf{E}_\perp^{\text{out}}(\mathbf{x}_\perp, z)$, we may write

$$\begin{aligned}\mathbf{H}_\perp^{\text{out}}(\mathbf{x}_\perp, 0) &= \sum_n a_n \frac{1}{Z_n}\hat{\mathbf{z}} \times \mathbf{u}_n(\mathbf{x}_\perp)\\ &= \frac{1}{2}\sum_n (a_n + b_n)\frac{1}{Z_n}\hat{\mathbf{z}} \times \mathbf{u}_n(\mathbf{x}_\perp)\\ &\quad + \frac{1}{2}\sum_n \frac{1}{Z_n}(a_n - b_n)\hat{\mathbf{z}} \times \mathbf{u}_n(\mathbf{x}_\perp)\\ &= \frac{1}{2}\mathbf{H}_\perp(\mathbf{x}_\perp, 0) + \frac{1}{2}\int_W d\mathbf{x}'_\perp\, \overleftrightarrow{\tilde{\mathbf{H}}}(\mathbf{x}_\perp, \mathbf{x}'_\perp)\\ &\quad \cdot (\hat{\mathbf{z}} \times \mathbf{E}_\perp(\mathbf{x}'_\perp, 0))\end{aligned} \qquad (86)$$

where the dyad

$$\overleftrightarrow{\tilde{\mathbf{H}}}(\mathbf{x}_\perp, \mathbf{x}'_\perp) = \sum_n \frac{1}{Z_n}(\hat{\mathbf{z}} \times \mathbf{u}_n(\mathbf{x}_\perp))(\hat{\mathbf{z}} \times \mathbf{u}_n(\mathbf{x}'_\perp)) \qquad (87)$$

is the analogue of the scalar distribution $\tilde{H}(\mathbf{x}_\perp, \mathbf{x}'_\perp)$. Dropping the spatial coordinate z, we get an alternative form of the waveguide integral equation

$$\begin{aligned}2\mathbf{H}_\perp^{\text{out}}(\mathbf{x}_\perp) = \mathbf{H}_\perp(\mathbf{x}_\perp) + \int_W d\mathbf{x}'_\perp\, \overleftrightarrow{\tilde{\mathbf{H}}}(\mathbf{x}_\perp, \mathbf{x}'_\perp)\\ \cdot (\hat{\mathbf{z}} \times \mathbf{E}_\perp(\mathbf{x}'_\perp))\end{aligned} \qquad (88)$$

or in terms of equivalent currents

$$\begin{aligned}2\mathbf{H}_\perp^{\text{out}}(\mathbf{x}_\perp) = -\hat{\mathbf{z}} \times \mathbf{J}_\perp(\mathbf{x}_\perp) - \int_W d\mathbf{x}'_\perp\, \overleftrightarrow{\tilde{\mathbf{H}}}(\mathbf{x}_\perp, \mathbf{x}'_\perp)\\ \cdot \mathbf{M}_\perp(\mathbf{x}'_\perp).\end{aligned} \qquad (89)$$

Equations (85) and (89) are the electromagnetic counterparts of the scalar waveguide integral equations given in (12) and (18).

D. Discretization

As stated above, the TM and TE vector modes on W are derivable from the scalar modes on W with Dirichlet and Neumann boundary conditions on ∂W, respectively, and the TEM vector modes (if any) are derivable from the solutions to the electrostatic potential problem on W. One can compute approximate solutions for the scalar modes and the electrostatic potential by putting scalar basis functions on W and following the procedure given in the Appendix. Once this has been accomplished, one has to choose between keeping the representation of the modes in terms of the scalar discretization or converting it to an equivalent vector discretization. If the scalar discretization is kept on the aperture, specialized code must be written to handle interactions with the waveguide aperture. On the other hand, if the waveguide modes are converted to a vector discretization early on, then the interactions between the various scattering surfaces, whether physical or waveguide aperture, can be handled in a consistent fashion, i.e., entirely in terms of vector basis functions. For computations involving more than just the waveguide alone, we find the later choice to be the simplest and cleanest to implement.

If we discretize the electric current $\mathbf{J}(\mathbf{x}_\perp)$ and magnetic current $\mathbf{M}(\mathbf{x}_\perp)$ on W in terms of M vector basis functions $\mathbf{f}_m(\mathbf{x}_\perp)$ using

$$\mathbf{J}(\mathbf{x}_\perp) \approx \sum_{m=1}^{M} I_m^W \mathbf{f}_m(\mathbf{x}_\perp) \tag{90}$$

$$\mathbf{M}(\mathbf{x}_\perp) \approx \sum_{m=1}^{M} S_m^W (\mathbf{f}_m(\mathbf{x}_\perp) \times \hat{\mathbf{z}}) \tag{91}$$

we may write the first waveguide integral equation (85) in its discretized form as

$$2V^W = N^W S^W - X^W I^W \tag{92}$$

where

$$V_i^W = \int_W d\mathbf{x}_\perp \mathbf{E}_\perp^{\text{out}}(\mathbf{x}_\perp) \cdot \mathbf{f}_i(\mathbf{x}_\perp) \tag{93a}$$

$$N_{ij}^W = \int_W d\mathbf{x}_\perp \mathbf{f}_i(\mathbf{x}_\perp) \cdot \mathbf{f}_j(\mathbf{x}_\perp) \tag{93b}$$

$$\begin{aligned} X_{ij}^W &= \int_W d\mathbf{x}_\perp \int_W d\mathbf{x}'_\perp \mathbf{f}_i(\mathbf{x}_\perp) \cdot \overline{\mathbf{H}}(\mathbf{x}_\perp, \mathbf{x}'_\perp) \cdot \mathbf{f}_j(\mathbf{x}'_\perp) \\ &= [(BN^W)^T \Lambda (BN^W)]_{ij} \end{aligned} \tag{93c}$$

and

$$\mathbf{u}_m(\mathbf{x}) = \sum_n B_{mn} \mathbf{f}_n(\mathbf{x}) \tag{94}$$

$$\Lambda_{mn} = Z_n \delta_{mn}. \tag{95}$$

We get the elements of B_{mn} by computing inner products of the vector basis functions with gradients of the scalar basis functions. For example, if $\mathbf{u}_m$ corresponds to a TM mode, it is clear from (32), (77), and (94) and the definition of N^W that the entries in the mth row of B_{mn} are given by

$$\begin{aligned} B_{mn} = \frac{1}{\sqrt{k^2 - \beta_m^2}} \sum_{jk} A_{mj} \int_W d\mathbf{x}_\perp \nabla_\perp f_j(\mathbf{x}_\perp) \\ \cdot \mathbf{f}_k(\mathbf{x}_\perp)((N^W)^{-1})_{kn}. \end{aligned} \tag{96}$$

Similarly, the discretized form of the second waveguide integral equation (89) becomes

$$2\tilde{V}^W = N^W I^W - \tilde{X}^W S^W \tag{97}$$

where

$$\tilde{V}_i^W = \int_W d\mathbf{x}_\perp \mathbf{H}_\perp^{\text{out}}(\mathbf{x}_\perp) \cdot (\mathbf{f}_i(\mathbf{x}_\perp) \times \hat{\mathbf{z}}) \tag{98a}$$

$$\begin{aligned} \tilde{X}_{ij}^W &= \int_W d\mathbf{x}_\perp \int_W d\mathbf{x}'_\perp (\mathbf{f}_i(\mathbf{x}_\perp) \times \hat{\mathbf{z}}) \cdot \overline{\tilde{\mathbf{H}}}(\mathbf{x}_\perp, \mathbf{x}'_\perp) \\ &\quad \cdot (\mathbf{f}_j(\mathbf{x}'_\perp) \times \hat{\mathbf{z}}) \\ &= [(BN^W)^T \tilde{\Lambda} (BN^W)]_{ij} \end{aligned} \tag{98b}$$

and

$$\tilde{\Lambda}_{mn} = \frac{1}{Z_n} \delta_{mn} = (\Lambda^{-1})_{mn}. \tag{99}$$

E. Coupled Integral Equations in the Perfect Conductor Case

Suppose the waveguide W is the primary source of radiation for a general antenna problem in which all other scattering surfaces S may be treated as perfect conductors. If there are no other sources, the electric field integral equation (EFIE) for $\mathbf{x}$ on $S \mp W$ is [7]

$$\begin{aligned} 0 = -\frac{1}{2}\hat{\mathbf{n}}(\mathbf{x}) \times \mathbf{M}(\mathbf{x}) + \oint_{S \mp W} ds' \Bigg[i\omega\mu \left(\overline{\mathbf{1}} + \frac{1}{k^2}\nabla'\nabla' \right) \\ \times G(\mathbf{x}, \mathbf{x}') \cdot \mathbf{J}(\mathbf{x}') + \nabla' G(\mathbf{x}, \mathbf{x}') \times \mathbf{M}(\mathbf{x}') \Bigg]_{\tan}. \end{aligned} \tag{100}$$

The tangential component of the electric field vanishes on a perfect conductor; hence, $\mathbf{M} = 0$ on S. At this point, we could rewrite the above equation in the separate forms appropriate to $\mathbf{x}$ on S and $\mathbf{x}$ on W and eliminate $\mathbf{M}$ on W by means of (85), thereby obtaining a set of coupled integral equations for the fields on S and W, just as we did in the scalar case. Then we could convert them to discretized form. Alternatively, we could discretize (100) as it stands, eliminate the unknown equivalent magnetic current amplitudes on W using (92) and achieve the discretized form directly. For brevity, we follow the latter approach.

A discretized version of (100) in block matrix form is

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} Z^{SS} & Z^{SW} & Y^{SW} \\ Z^{WS} & Z^{WW} & -\frac{1}{2}N^W \end{bmatrix} \begin{bmatrix} I^S \\ I^W \\ S^W \end{bmatrix} \tag{101}$$

where

$$\begin{aligned} Z_{ij}^{\alpha\beta} = i\omega\mu \int_\alpha ds \int_\beta ds' \mathbf{f}_i^\alpha(\mathbf{x}) \cdot \left(\overline{\mathbf{1}} + \frac{1}{k^2}\nabla'\nabla' \right) G(\mathbf{x}, \mathbf{x}') \\ \cdot \mathbf{f}_j^\beta(\mathbf{x}') \end{aligned} \tag{102}$$

$$Y_{ij}^{\alpha\beta} = \int_\alpha ds \int_\beta ds' \mathbf{f}_i^\alpha(\mathbf{x}) \cdot (\nabla' G(\mathbf{x}, \mathbf{x}') \times (\mathbf{f}_j^\beta(\mathbf{x}') \times \hat{\mathbf{n}}')) \tag{103}$$

with S or W replacing α and β and I^S representing the block of unknown current amplitudes on S, which is related to the electric current $\mathbf{J}$ on S by

$$\mathbf{J}(\mathbf{x}) \approx \sum_m I_m^S \mathbf{f}_m(\mathbf{x}). \tag{104}$$

Rewriting (92) as

$$S^W = 2(N^W)^{-1}V^W + (N^W)^{-1}X^W I^W \tag{105}$$

we can eliminate the block of unknowns S^W in favor of I^W to obtain the discretized version of (100) in its simplest block form

$$\begin{bmatrix} -2Y^{SW}(X^{W})^{-1}V^{W} \\ V^{W} \end{bmatrix} = \begin{bmatrix} Z^{SS} & Z^{SW} + Y^{SW}(X^{W})^{-1} \\ Z^{WS} & Z^{WW} - \frac{1}{2}X^{W} \end{bmatrix} \begin{bmatrix} I^{S} \\ I^{W} \end{bmatrix}. \tag{106}$$

F. Modal Decomposition

By employing the first completeness relation for the modes, we can decompose the transverse part of the electric field into a sum over modes as

$$\mathbf{E}_{\perp}(\mathbf{x}) = \sum_n \eta_n \mathbf{u}_n(\mathbf{x}) \tag{107}$$

where

$$\eta_n = \int_W ds\, \mathbf{u}_n(\mathbf{x}) \cdot \mathbf{E}_{\perp}(\mathbf{x}) \tag{108}$$

is the amplitude of the nth mode contained in $\mathbf{E}_{\perp}(\mathbf{x})$. It is useful to further decompose $\mathbf{E}_{\perp}(\mathbf{x})$ into its incoming and outgoing components

$$\mathbf{E}_{\perp}(\mathbf{x}) = \mathbf{E}^{\text{in}}_{\perp}(\mathbf{x}) + \mathbf{E}^{\text{out}}_{\perp}(\mathbf{x}). \tag{109}$$

Since the discretization of $\mathbf{E}^{\text{out}}_{\perp}(\mathbf{x})$ is given by V^W, we may write the discretized form of η^{out}_n as

$$\eta^{\text{out}}_n = \sum_m A_{nm} V^W_m. \tag{110}$$

Using (85) to eliminate $\mathbf{E}_{\perp}(\mathbf{x})$, we arrive at the discretized form of η^{in}_n

$$\eta^{\text{in}}_n = \sum_m A_{nm}(V^W + X^W I^W)_m. \tag{111}$$

Similarly, by employing the second completeness relation for the modes, we may decompose the transverse part of the magnetic field as

$$\mathbf{H}_{\perp}(\mathbf{x}) = \sum_n \tilde{\eta}_n(\hat{\mathbf{z}} \times \mathbf{u}_n(\mathbf{x})) \tag{112}$$

where

$$\tilde{\eta}_n = \int_W ds(\hat{\mathbf{z}} \times \mathbf{u}_n(\mathbf{x})) \cdot \mathbf{H}_{\perp}(\mathbf{x}). \tag{113}$$

Then, using

$$\mathbf{H}_{\perp}(\mathbf{x}) = \mathbf{H}^{\text{in}}_{\perp}(\mathbf{x}) + \mathbf{H}^{\text{out}}_{\perp}(\mathbf{x}) \tag{114}$$

and (89), we can write $\tilde{\eta}^{\text{out}}_n$ and $\tilde{\eta}^{\text{in}}_n$ in discretized form as

$$\tilde{\eta}^{\text{out}}_n = -\sum_m A_{nm} \tilde{V}^W_m \tag{115}$$

and

$$\tilde{\eta}^{\text{in}}_n = -\sum_m A_{nm}(\tilde{V}^W + \tilde{X}^W S^W)_m. \tag{116}$$

G. Power

The time-averaged power-flow-density vector (Poynting vector) is [6]

$$\langle \mathbf{S}(\mathbf{x}) \rangle = \frac{1}{2} \text{Re}[\mathbf{E}_{\perp}(\mathbf{x}) \times \mathbf{H}_{\perp}(\mathbf{x})^*]. \tag{117}$$

The total power flowing across the waveguide aperture in the $\hat{\mathbf{z}}$ direction is made up of an incoming part associated with the incoming parts of $\mathbf{E}_{\perp}$ and $\mathbf{H}_{\perp}$ and an outgoing part associated with the outgoing parts of $\mathbf{E}_{\perp}$ and $\mathbf{H}_{\perp}$. The total power exiting (entering) the waveguide aperture is given by

$$\begin{aligned} P^{\alpha} &= \int_W ds \langle \mathbf{S}^{\alpha}(\mathbf{x}) \rangle \cdot \hat{\mathbf{z}} \\ &= \frac{1}{2} \int_W ds\, \text{Re}[\mathbf{E}^{\alpha}_{\perp}(\mathbf{x}) \times \mathbf{H}^{\alpha}_{\perp}(\mathbf{x})^*] \end{aligned} \tag{118}$$

for α = out (in). This integral is most conveniently evaluated by decomposing $\mathbf{E}^{\alpha}_{\perp}$ and $\mathbf{H}^{\alpha}_{\perp}$ into their modal components, since the modes are orthogonal and the power in the sum over modes is equal to the sum of the powers in each mode.

The amplitude of the nth outgoing (incoming) mode contained in $\mathbf{E}_{\perp}(\mathbf{x})$ is $\eta^{\text{out}}_n(\eta^{\text{in}}_n)$. Therefore, the time-averaged power exiting (entering) the waveguide aperture is

$$P^{\alpha} = \sum_n^{n_{\max}} \frac{|\eta^{\alpha}_n|^2}{2Z_n} \tag{119}$$

for α = out (in) where $n_{\max}$ is the largest value of n for which β_n is real. We exclude modes with imaginary propagation constants since such modes do not transport any power into or out of the guide on average.

The amplitude of the nth outgoing (incoming) mode contained in $\mathbf{H}_{\perp}$ is $\tilde{\eta}^{\text{out}}_n$ $(\tilde{\eta}^{\text{in}}_n)$. Therefore, the time-averaged power exiting (entering) the waveguide aperture is

$$P^{\alpha} = \sum_n^{n_{\max}} \frac{Z_n |\tilde{\eta}^{\alpha}_n|^2}{2} \tag{120}$$

for α = out (in).

IV. EXTENSIONS

Up to this point, we have assumed that all energy coupled into incoming traveling modes is completely absorbed. It is possible (at the cost of some extra complication) to relax this assumption, as we now demonstrate for scalar scattering.

Suppose a uniform waveguide is terminated after length L by a wall (oriented perpendicular to the axis of the guide) whose reflectivity for the mth waveguide mode is r_m. For the time being, assume no independent sources are located inside the guide. Every mode that enters with amplitude b_n, exits with amplitude $a_n = r_n e^{i\beta_n 2L} b_n$, i.e., if $\psi^{\text{in}}(\mathbf{x}) = \sum_n b_n u_n(\mathbf{x})$ comes in, then $\psi^{\text{out}}(\mathbf{x}) = \sum_n r_n e^{i\beta_n 2L} b_n u_n(\mathbf{x})$ goes out. This expression for $\psi^{\text{out}}(\mathbf{x})$ can be rewritten as

$$\psi^{\text{out}}(\mathbf{x}) = \int_W ds' R(\mathbf{x}, \mathbf{x}') \psi^{\text{in}}(\mathbf{x}') \tag{121}$$

where

$$R(\mathbf{x}, \mathbf{x}') = \sum_n r_n e^{i\beta_n 2L} u_n(\mathbf{x}) u_n(\mathbf{x}'). \tag{122}$$

After discretization, (121) becomes

$$N^{W} S^{\text{out}} = (AN^{W})^{T} R (AN^{W}) S^{\text{in}} \tag{123}$$

where R is a diagonal reflectivity matrix whose elements are

$$R_{ij} = r_i e^{i\beta_i 2L} \delta_{ij}. \tag{124}$$

A boundary condition relating ψ and σ on W can be obtained by applying the operator $\int_W ds'(\delta(\mathbf{x}, \mathbf{x}') + R(\mathbf{x}, \mathbf{x}'))\cdot$ to both sides of (12) and using (52). The result is

$$\int_W ds'(\delta(\mathbf{x} - \mathbf{x}') - R(\mathbf{x}, \mathbf{x}')) \int_W ds'' H(\mathbf{x}', \mathbf{x}'')\sigma(\mathbf{x}'')$$
$$= \int_W ds'(\delta(\mathbf{x}, \mathbf{x}') - R(\mathbf{x}, \mathbf{x}'))\psi(\mathbf{x}') \tag{125}$$

or in discretized form

$$(N^{W} - R)(N^{W})^{-1} X^{W} I^{W} = (N^{W} - R) S^{W}. \tag{126}$$

The discretized relation takes a particularly simple and appealing form if: 1) the basis functions used on W are orthonormal in which case $N^{W} = 1$ and 2) if as many modes are computed as there are basis functions on W in which case $A^T A = 1$. Then (126) is equivalent to

$$T X^{W} I^{W} = S^{W} \tag{127}$$

where

$$T_{ij} = [A^T t A]_{ij} \tag{128}$$

and

$$t_{mn} = \frac{1 + r_m}{1 - r_m} \delta_{mn} \tag{129}$$

is the diagonal transmission matrix giving the amplitude transmission of each mode at the waveguide aperture.

It is easy to modify these relations to allow for a specified outgoing wave. Suppose the field $\psi^{\text{spec}}(\mathbf{x})$ is specified as being emitted from the aperture in addition to the reflected wave, i.e., $\psi^{\text{out}}(\mathbf{x}) = \psi^{\text{spec}}(\mathbf{x}) + \psi^{\text{refl}}(\mathbf{x})$. We use $\psi^{\text{refl}}(\mathbf{x})$ here to refer to the quantity on the left side of (121). The result is

$$\int_W ds'(\delta(\mathbf{x} - \mathbf{x}') + R(\mathbf{x}, \mathbf{x}')) \int_W ds'' H(\mathbf{x}', \mathbf{x}'')\sigma(\mathbf{x}'')$$
$$= \int_W ds'(\delta(\mathbf{x}, \mathbf{x}') - R(\mathbf{x}, \mathbf{x}'))\psi(\mathbf{x}') - 2\psi^{\text{spec}}(\mathbf{x}). \tag{130}$$

Its discretized form

$$2V^{\text{spec}} = (N^{W} - R) S^{W} - (N^{W} + R)(N^{W})^{-1} X^{W} I^{W} \tag{131}$$

is the obvious analog to (36) and reduces to it for R — 0.

Even more generally, one can imagine the situation in which each incoming mode can be scattered into one or more outgoing modes. Any number of practical effects (such as nonuniformities in the cross section or imperfect termination) could cause this to happen. In such a case, the reflectivity matrix R contains the amplitude for every mode to scatter into every other mode and is no longer diagonal.

Analogous results obtain for the alternative form of the scalar waveguide boundary condition and for the vector cases.

V. SUMMARY

As the previous discussion illustrates, the equations that describe scattering interactions with waveguides can be put into simple forms that are common to scalar scattering and vector scattering. For example, the boundary condition on a waveguide aperture may be written in both cases as

$$2V^{W} = N^{W} S^{W} - X^{W} I^{W} \tag{132}$$

or

$$2\tilde{V}^{W} = N^{W} I^{W} - \tilde{X}^{W} S^{W}. \tag{133}$$

In the scalar case, the unknown amplitudes I^{W} and S^{W} are related to the field ψ and its longitudinal derivative σ according to (33) and (34): the matrices N^{W}, X^{W}, and $\tilde{X}^{W}$ and the vectors V^{W} and $\tilde{V}^{W}$ are given by (37) and (40). In the vector case, the unknown amplitudes I^{W} and S^{W} are related to the equivalent electric and magnetic currents J and M, according to (90) and (91): the matrices N^{W}, X^{W}, and $\tilde{X}^{W}$ and the vectors V^{W} and $\tilde{V}^{W}$ are given by (93) and (98). The discretized equations for scalar scattering when W obeys the waveguide boundary condition and S obeys Dirichlet boundary conditions [see (43)] are also identical to the equations for vector scattering when W obeys the waveguide boundary condition and S is perfectly conducting [see (106)]. The commonality extends to the expressions for power transport into and out of the waveguide as well.

APPENDIX

Construction of the X and $\tilde{X}$ matrices that appear in the discretized expressions for the waveguide boundary condition requires an approximate representation of the eigenmodes in terms of basis functions on patches covering the waveguide aperture as well as the eigenvalues associated with these eigenmodes. For a few geometries such as rectangular waveguide and coaxial waveguide, complete analytical solutions for the eigenmodes are known. In such cases, it is a simple matter to calculate the projection of a given eigenmode onto the set of basis functions. In the general case, an eigenvalue equation must be constructed for computing the modes.

In this Appendix we describe a means for computing the modes of cylindrical waveguides of arbitrary cross section. There are three subsections. The first and second subsections describe methods for numerically solving the scalar Helmholtz equation for the waveguide modes when the waveguide walls obey either Dirichlet or Neumann boundary conditions, respectively. The third subsection describes a method for numerically solving the scalar Laplace equation for the electrostatic potential of a multiply-connected cylindrical waveguide, all but one of whose surfaces is held at zero potential.

The Helmholtz modes are directly applicable to scalar problems such as acoustic radiation and scattering. The Helmholtz and Laplace modes are applicable to electromagnetic radiation and scattering problems in that the TM and TE modes can be deduced from the scalar Helmholtz modes with Dirichlet and Neumann boundary conditions, respectively, and the TEM modes are derivable from the scalar Laplace modes. The correspondence is described further in Section III-B of the main text.

We will assume the availability of scalar basis functions that are continuous across patch boundaries. A simple example of such a basis function is a function that spans two triangular patches sharing a common edge and whose value goes linearly from unity on the common edge to zero at the opposing vertices. The extension of continuous scalar basis functions to higher order polynomials in the surface parameterization results in three types of basis functions that may be classified according to whether they span two patches that share a common edge, span multiple patches that share a common vertex, or have single patch support. Basis functions of the first variety go to zero at the opposing vertices and are nonzero on the common edge; basis functions of the second variety go to zero on all edges not touching the central vertex (where they are nonzero); basis functions of the third variety are zero on the boundary of a patch and nonzero in its interior.

A. Scalar Helmholtz Modes

1) Dirichlet Boundary Conditions on ∂W: Operating on both sides of (5) by $\int_W d\mathbf{x}_\perp f_m(\mathbf{x}_\perp)$ turns it into an integral equation, which may be written as

$$-\int_W d\mathbf{x}_\perp f_m(\mathbf{x}_\perp)(\nabla_\perp\cdot\nabla_\perp u_n(\mathbf{x}_\perp)) = (k^2-\beta_n^2)\int_W d\mathbf{x}_\perp f_m(\mathbf{x}_\perp)u_n(\mathbf{x}_\perp). \tag{134}$$

Integrating the left-hand side by parts and applying Gauss' theorem to convert one of the resulting surface integrals into a boundary integral, we get

$$\int_W d\mathbf{x}_\perp \nabla_\perp f_m(\mathbf{x}_\perp)\cdot\nabla_\perp u_n(\mathbf{x}_\perp) - \oint_{\partial W} dl f_m(\mathbf{x}_\perp)(\hat{\mathbf{e}}_\perp(\mathbf{x}_\perp)\cdot\nabla_\perp u_n(\mathbf{x}_\perp)) = (k^2-\beta_n^2)\int_W d\mathbf{x}_\perp f_m(\mathbf{x}_\perp)u_n(\mathbf{x}_\perp) \tag{135}$$

where $\hat{\mathbf{e}}_\perp(\mathbf{x}_\perp)$ is the unit edge normal to ∂W at $\mathbf{x}_\perp$. The unit edge normal is in the plane of W and points into the waveguide wall.

The Dirichlet boundary condition demands that $u_n(\mathbf{x}_\perp\in\partial W)=0$. If we expand the modes u_n in a set of basis functions f_m that are continuous and vanish on the boundary of W, i.e.,

$$u_n(\mathbf{x}_\perp)=\sum_m A_{nm}f_m(\mathbf{x}_\perp) \tag{136}$$

then the boundary integral term vanishes and (135) becomes a generalized eigenvalue equation for the mode coefficients

$$\sum_{m'} M_{mm'}A_{nm'} = (k^2-\beta_n^2)\sum_{m'} N_{mm'}A_{nm'} \tag{137}$$

where

$$N_{mm'} \equiv \int_W d\mathbf{x}_\perp f_m(\mathbf{x}_\perp)f_{m'}(\mathbf{x}_\perp) \tag{138}$$

$$M_{mm'} \equiv \int_W dx_\perp \nabla_\perp f_m(\mathbf{x}_\perp)\cdot\nabla_\perp f_{m'}(\mathbf{x}_\perp). \tag{139}$$

2) Neumann Boundary Conditions on ∂W: The Neumann boundary condition demands that $(\hat{\mathbf{e}}_\perp\cdot\nabla_\perp)u_n(\mathbf{x}_\perp\in\partial W)=0$. If we had basis functions whose values were nonzero on the boundary but whose edge derivatives vanished on the boundary, we could construct the modes directly from them, just as we did in the Dirichlet case. Since we do not, we need to augment our usual set of basis functions on the interior of W with extra basis functions associated with the boundary of W. Edge-based basis functions supported on the patch pairs (one each from S and W) that share a common edge on ∂W comprise this set.

The generalized eigenvalue equation again derives from (135) and (136). In this case, however, the unknown coefficients A_{nm} also need to obey the added constraint that the edge derivative of each eigenmode must vanish on the boundary. We may write this constraint in integral form as

$$\oint_{\partial W} dl\,\hat{\mathbf{e}}_\perp(\mathbf{x}_\perp)\cdot\nabla_\perp u_n(\mathbf{x}_\perp)=0 \tag{140}$$

which, after substituting the discretized approximation for u_n, becomes

$$\sum_m C_m A_{nm}=0 \tag{141}$$

where

$$C_m \equiv \int_{\partial W} dl\,\hat{\mathbf{e}}_\perp(\mathbf{x}_\perp)\cdot\nabla_\perp f_m(\mathbf{x}_\perp). \tag{142}$$

Thus, we seek solutions to the eigenvalue equation

$$\sum_{m'}(M_{mm'}-L_{mm'})A_{nm'} = (k^2-\beta_n^2)\sum_{m'} N_{mm'}A_{nm'} \tag{143}$$

where

$$L_{mm'} \equiv \oint_{\partial W} dl f_m(\mathbf{x}_\perp)(\hat{\mathbf{e}}_\perp(\mathbf{x}_\perp)\cdot\nabla_\perp f_{m'}(\mathbf{x}_\perp)) \tag{144}$$

and the matrices M and N are defined as in the Dirichlet case, subject to the constraint given by (141).

We can subsume the constraint information directly into the eigenvalue equation by use of the projection operator P defined by

$$P \equiv 1 - C^T(CC^T)^{-1}C \tag{145}$$

where C is given above and 1 represents the identity matrix of the proper dimensionality. P has the property that it reproduces vectors x that obey $Cx=0$ and it annihilates vectors that do not. P also has the property that the vectors x that simultaneously obey the eigenvalue equation $Qx=\lambda x$ and the constraint equation $Cx=0$, are the same vectors that obey the eigenvalue equation

$$PQPx=\lambda x. \tag{146}$$

Applying this to (143), we obtain the following the generalized eigenvalue equation for Neumann boundary conditions:

$$\sum_{m'}\left[PN^{-1}(M-L)P\right]_{mm'}A_{nm'} = (k^2-\beta_n^2)A_{nm}. \tag{147}$$

Rows of A (i.e., eigenvectors) corresponding to eigenmodes that do not obey the constraint will vanish (to numerical precision) when left multiplied by P. All such eigenmodes and eigenvectors should be discarded.

B. Scalar Laplace Modes

We seek solutions $u_n(\mathbf{x}_\perp)$ that obey the Laplace equation

$$\nabla_\perp^2 u_n(\mathbf{x}_\perp) = 0 \tag{148}$$

inside W and vanish on all boundaries of W except one (call it ∂W_n), where we may arbitrarily set it to unity. Since our basis functions vanish on the boundary, we need to construct a special function $v_n(\mathbf{x}_\perp)$ that is continuous and evaluates to unity on ∂W_n. For example, given triangular patches parameterized by the three (nonindependent) triangle coordinates u_1, u_2, and u_3, we could take $v_n = 0$ on all patches that are not in contact with the boundary, $v_n = u_i$ on all patches that have the vertex $u_i = 1$ on the boundary, and $v_n = 1 - u_i$ on all patches that have edge $u_i = 0$ on the boundary. Then we want to approximately solve

$$\nabla_\perp^2\left(v_n(\mathbf{x}_\perp) - \sum_{m'} A_{nm'} f_{m'}(\mathbf{x}_\perp)\right) = 0. \tag{149}$$

Applying the operator $\int_W d\mathbf{x}_\perp f_m(\mathbf{x}_\perp)$ to both sides and integrating the resulting equation by parts produces the following linear equation for the basis function coefficients $A_{nm'}$ for the potential function associated with the nth boundary:

$$\sum_{m'} M_{mm'} A_{nm'} = \int_W d\mathbf{x}_\perp \nabla_\perp f_m(x_\perp) \cdot \nabla_\perp v_n(x_\perp) \tag{150}$$

where M is as defined in (139).

To make normalized TEM modes out of these Laplace modes, we need them to obey

$$\begin{aligned}
1 &= \int_W d\mathbf{x}_\perp \mathbf{u}_n(x_\perp) \cdot \mathbf{u}_n(x_\perp) \\
&= \int_W d\mathbf{x}_\perp \nabla_\perp u_n(x_\perp) \cdot \nabla_\perp u_n(x_\perp) \\
&= \int_W d\mathbf{x}_\perp \nabla_\perp \cdot (u_n(x_\perp)\nabla_\perp u_n(x_\perp)) \\
&\quad - \int_W d\mathbf{x}_\perp \nabla_\perp^2 u_n(x_\perp) \\
&= \oint_{\partial W} dl u_n(\mathbf{x}_\perp)(\hat{\mathbf{e}}_\perp(\mathbf{x}_\perp) \cdot \nabla_\perp u_n(\mathbf{x}_\perp)) \\
&= \int_{\partial W_n} dl(\hat{\mathbf{e}}_\perp(\mathbf{x}_\perp) \cdot \nabla_\perp u_n(\mathbf{x}_\perp))
\end{aligned} \tag{151}$$

which means the coefficients of the discretized representation of u_n must be scaled to make

$$\begin{aligned}
1 &= \int_{\partial W_n} dl\hat{\mathbf{e}}_\perp(\mathbf{x}_\perp) \cdot \nabla_\perp\left(\sum_{m'} A_{nm'} f_{m'}(\mathbf{x}_\perp)\right) \\
&= \sum_{m'} A_{nm'} \int_{\partial W_n} dl\hat{\mathbf{e}}_\perp(\mathbf{x}_\perp) \cdot \nabla_\perp f_{m'}(\mathbf{x}_\perp).
\end{aligned} \tag{152}$$

REFERENCES

[1] R. F. Harrington, *Field Computation by Moment Methods*. New York: Macmillan, 1968.

[2] D. T. McGrath and V. P. Pyati, "Phased array antenna analysis with the hybrid finite element method," *IEEE Trans. Antennas Propagat.*, vol. 42, pp. 1625–1630, Dec. 1994.

[3] D. S. Jones, *The Theory of Electromagnetism*. New York: Pergamon, 1964.

[4] A. W. Maue, "Toward formulation of a general diffraction problem via an integral equation," *Zeitschrift Phys.*, vol. 126, pp. 601–618, 1949.

[5] P. M. Morse and H. Feshbach, *Methods of Theoretical Physics*. New York: McGraw-Hill, 1953.

[6] J. D. Jackson, *Classical Electrodynamics*, 2nd ed. New York: Wiley, 1975.

[7] N. Morita, N. Kumagai, and J. R. Mautz, *Integral Equation Methods for Electromagnetics*. Norwood, MA: Artech House, 1990.

John J. Ottusch was born in Landstuhl, West Germany, in 1955. He received the B.S. degree from the Massachusetts Institute of Technology, Cambridge, in 1977, and the Ph.D. degree from the University of California, Berkeley, in 1985, both in physics.

His doctoral thesis research involved heterodyne spectroscopy of the sun in the mid-infrared. Since 1985, he has been a member of the Technical Staff at Hughes Research Laboratories (now HRL Laboratories), Malibu, CA. From 1985 to 1994 his research was primarily devoted to experimental investigations of nonlinear optics, including stimulated Raman and Brillouin scattering and optical phase conjugation. Since 1994 he has been working on developing fast, high-order algorithms and software for electromagnetic modeling.

Dr. Ottusch received a National Science Foundation fellowship at the University of California, Berkeley, in 1985.

George C. Valley was born in Winchester, MA, in 1944. He received the B.A. degree in physics from Dartmouth College, Hanover, NH, in 1966, and the Ph.D. degree in physics from the University of Chicago, Chicago, IL, in 1971.

He worked at Cornell Aeronautical Laboratory (now Calspan Corp.) from 1972 to 1977 on RF and optical propagation. He joined Hughes Aircraft Company (now Hughes Electronics), Los Angeles, CA, in 1977. He worked on high-energy laser propagation and adaptive optics until 1980, when he transferred to Hughes Research Laboratories, Malibu, CA, where he worked on nonlinear optical-phase conjugation, nonlinear optics in photorefractive materials, visible lasers for displays, spatial solitons and electromagnetic modeling. He recently transferred to Hughes Space and Communications Company in El Segundo, CA, to work on optical intersatellite links. His research interests include nonlinear optics, laser physics, optical communication, and electromagnetic modeling.

Dr. Valley is a member of the American Physical Society and a Fellow of the Optical Society of America.

Stephen Wandzura received the B.S. degree in music from University of California, Los Angeles, in 1971, and the Ph.D. degree in physics from Princeton University, Princeton, NJ, in 1977.

He is a Principal Research Scientist at the Communications and Photonics Laboratory, HRL Laboratories, Malibu, CA. His research has been in diverse areas of scattering and propagation. His thesis research studied spin-dependent deep inelastic lepton-hadron scattering. As a National Research Fellow with the NOAA, he studied scattering of light by atmospheric turbulence and the occurrence of mountain lee waves. At HRL, he has worked on classical and quantum optics, especially the theoretical and numerical study of stimulated scattering. Since 1989, he has be studying numerical solution of scattering and radiation problems. He has published articles in *Physical Review*, *Physical Review Letters*, *Physics Letters*, *Optics Letters*, *Nuclear Physics*, *JOSA*, and other journals.

IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION, VOL. 47, NO. 4, APRIL 1999

Numerical Solution of 2-D Scattering Problems Using High-Order Methods

Lisa R. Hamilton, John J. Ottusch, Mark A. Stalzer, R. Steven Turley, *Senior Member, IEEE*,
John L. Visher, and Stephen M. Wandzura

Abstract—**We demonstrate that a method of moments scattering code employing high-order methods can compute accurate values for the scattering cross section of a smooth body more efficiently than a scattering code employing standard low-order methods. Use of a high-order code also makes it practical to provide meaningful accuracy estimates for computed solutions.**

Index Terms— **Boundary integral equation, electromagnetic scattering, high-order numerical method, method of moments.**

I. INTRODUCTION

A common misconception about method of moments solutions to scattering problems is that they cannot produce results accurate to more than a few decimal places. Such a limitation cannot be fundamental. The method of moments technique results from discretizing an integral formulation of the wave equation, which, in its continuous form, is exact. We expect that the solution to the discretized integral equation will converge to the solution of the continuous integral equation in the limit as the discretization scale size is reduced to zero, if finite precision effects are negligible.

The problem with achieving high accuracy is not a fundamental one but rather a practical one, and it stems from the almost universal use of low-order numerical methods in scattering codes. Low-order numerical methods, while simpler to implement, suffer from the fact that the computer resources (e.g., memory and CPU time) required to achieve a given solution accuracy grow rapidly as the accuracy requirement increases. Even for scatterers only a few wavelengths in size, the computer resources required to compute cross sections to more than a few digits of accuracy may be excessive. High-order methods are specifically designed to overcome such limitations by reducing the incremental cost of accuracy improvements.

FastScat^{TM} is a general purpose, method of moments scattering code [1] developed at Hughes Research Laboratories (now HRL Laboratories) that employs high-order methods in its current basis functions, quadratures, and geometry description. The focus of this paper is on the current basis functions and how they influence the convergence rate of computed cross sections for two dimensional (2-D) scattering problems. We will demonstrate that high-order methods make it practical to achieve solution accuracies limited only by machine precision. Such a demonstration is not merely of academic interest. High accuracies at intermediate stages of the calculation are sometimes required to achieve even engineering accuracies in the final result. Furthermore, the ability to obtain accuracy improvements at relatively low cost has the added benefit that it becomes possible to obtain meaningful estimates of the accuracy of a computed solution [2]. Without some estimate of its accuracy, a computed solution is of limited usefulness.

II. SCALAR INTEGRAL EQUATIONS

The electromagnetic scattering problem for a three-dimensional (3-D) scatterer that is translationally invariant in one direction can be decoupled into two independent problems, each of which is isomorphic to a two dimensional scalar scattering problem with a different boundary condition. In the TM case, the incident electric field is polarized parallel to the axis of symmetry; in the TE case, it is the incident magnetic field. The boundary conditions for the 2-D scalar scattering problem corresponding to a perfect electrical conductor (PEC) in 3-D are Dirichlet for TM polarization and Neumann for TE polarization.

For the TM polarization case $[\psi(\boldsymbol{x}' \text{ on } C) = 0]$, the electric field integral equation for PEC boundary conditions is

$$\phi^{\text{inc}}(\boldsymbol{x}) = -\oint_C dl'\, G(\boldsymbol{x},\boldsymbol{x}')\sigma(\boldsymbol{x}') \tag{1}$$

where ϕ^{inc} is the incident field and σ is the surface charge density. It is defined as the normal derivative of the total field ψ on the surface, i.e.,

$$\sigma(\boldsymbol{x}') \equiv -\hat{\boldsymbol{n}}' \cdot \nabla\psi(\boldsymbol{x}') \tag{2}$$

where $\hat{\boldsymbol{n}}'$ is the outward normal to the scattering surface at $\boldsymbol{x}'$. The integral is taken around the contour C given by the intersection of the 3-D scattering surface and a plane perpendicular to the axis of symmetry. The kernel G is the Green function of the Helmholtz wave equation in 2-D, namely

$$G(\boldsymbol{x},\boldsymbol{x}') = \frac{i}{4}H_0^{(1)}(k|\boldsymbol{x}-\boldsymbol{x}'|) \tag{3}$$

Manuscript received September 22, 1997; revised July 24, 1998. This work was supported by the Advanced Research Projects Agency of the U.S. Department of Defense and was monitored by the Air Force Office of Scientific Research under Contracts F49620-91-C-0064 and F49620-91-C-0084.

L. R. Hamilton, J. J. Ottusch, M. A. Stalzer, J. L. Visher, and S. M. Wandzura are with the Computational Physics Department of the Information Sciences Laboratory at HRL Laboratories, Malibu, CA 90265 USA.

R. S. Turley is with the Department of Physics and Astronomy at Brigham Young University, Provo, UT 84602 USA.

Publisher Item Identifier S 0018-926X(99)04775-4.

0018-926X/99$10.00 © 1999 IEEE

where $H_0^{(1)}$ is the zeroth-order Hankel function of the first kind and k is the wavenumber of the incident field. Similarly, for the TE polarization case [$\sigma(\boldsymbol{x}'$ on $C) = 0$], the electric field integral equation is

$$-\hat{\boldsymbol{n}} \cdot \nabla \psi^{\text{inc}}(\boldsymbol{x}) = (\hat{\boldsymbol{n}} \cdot \nabla) \oint_C dl' (\hat{\boldsymbol{n}}' \cdot \nabla' G(\boldsymbol{x}, \boldsymbol{x}')) \psi(\boldsymbol{x}'). \quad (4)$$

The correspondence between the scalar quantities ψ and σ and the parallel (to the surface) components of the electric and magnetic fields is given by

$$\boldsymbol{E}_{\parallel}(\boldsymbol{x}) = \psi(\boldsymbol{x})\hat{\boldsymbol{z}} \quad (5)$$

$$\boldsymbol{H}_{\parallel}(\boldsymbol{x}) = \frac{\sigma(\boldsymbol{x})}{i\omega\mu}\hat{\boldsymbol{z}} \times \hat{\boldsymbol{n}} \quad (6)$$

in the TM case and

$$\boldsymbol{H}_{\parallel}(\boldsymbol{x}) = \psi(\boldsymbol{x})\hat{\boldsymbol{z}} \quad (7)$$

$$\boldsymbol{E}_{\parallel}(\boldsymbol{x}) = \frac{\sigma(\boldsymbol{x})}{i\omega\epsilon}\hat{\boldsymbol{n}} \times \hat{\boldsymbol{z}} \quad (8)$$

in the TE case, where $\hat{\boldsymbol{z}}$ is the direction of translational invariance, $\hat{\boldsymbol{n}}$ is the surface normal, ω is the angular frequency, and ϵ and μ are the dielectric constant and magnetic susceptibility of the external medium, respectively. All fields implicitly contain the time dependence factor $e^{i\omega t}$.

A Galerkin method of moments solution [3] to the continuous scalar field equation, (1), proceeds by first expanding the unknown charge $\sigma(\boldsymbol{x})$ in terms of basis functions $f_j(\boldsymbol{x})$,

$$\sigma(\boldsymbol{x}) = \sum_j I_j f_j(\boldsymbol{x}) \quad (9)$$

and then testing the equation with each of the basis functions by applying the operator $\oint_C ds' f_i(\boldsymbol{x}')\cdot$ to both sides. The result is a matrix equation of the form

$$V = ZI \quad (10)$$

where

$$V_i = \oint_C dl\, \phi^{\text{inc}}(\boldsymbol{x}) f_i(\boldsymbol{x}) \quad (11)$$

and

$$Z_{ij} = \oint_C dl \oint_C dl'\, f_i(\boldsymbol{x}) G(\boldsymbol{x}, \boldsymbol{x}') f_j(\boldsymbol{x}'). \quad (12)$$

Similarly, we can discretize the scalar charge equation, (4), by expanding the unknown field as

$$\psi(\boldsymbol{x}) = \sum_j S_j f_j(\boldsymbol{x}) \quad (13)$$

and applying the testing operators to arrive at the matrix equation

$$\tilde{V} = \tilde{Z} S \quad (14)$$

where

$$\tilde{V}_i = -\oint_C dl\, [\hat{\boldsymbol{n}} \cdot \nabla \phi^{\text{inc}}(\boldsymbol{x})] f_i(\boldsymbol{x}) \quad (15)$$

and

$$\tilde{Z}_{ij} = \oint_C dl\, f_i(\boldsymbol{x})(\hat{\boldsymbol{n}} \cdot \nabla) \oint_C dl' (\hat{\boldsymbol{n}}' \cdot \nabla' G(\boldsymbol{x}, \boldsymbol{x}')) f_j(\boldsymbol{x}') \quad (16a)$$

$$= \oint_C dl \oint_C dl'\, f_i(\boldsymbol{x}) \cdot \left[\left(k^2(\hat{\boldsymbol{n}} \cdot \hat{\boldsymbol{n}}') - \frac{\partial^2}{\partial l \partial l'}\right) G(\boldsymbol{x}, \boldsymbol{x}')\right] f_j(\boldsymbol{x}') \quad (16b)$$

$$= \oint_C dl \oint_C dl' \cdot \left[k^2(\hat{\boldsymbol{n}} \cdot \hat{\boldsymbol{n}}') f_i(\boldsymbol{x}) f_j(\boldsymbol{x}') - \frac{\partial f_i(\boldsymbol{x})}{\partial l} \frac{\partial f_j(\boldsymbol{x}')}{\partial l'}\right] G(\boldsymbol{x}, \boldsymbol{x}'). \quad (16c)$$

The second form for $\tilde{Z}_{ij}$ is like the first in that it requires differentiating the kernel twice. In the first form they are normal derivatives; in the second they have been converted to tangential derivatives by use of the Helmholtz equation. Differentiating the kernel exacerbates the singularity of the kernel at $\boldsymbol{x} = \boldsymbol{x}'$, which is unattractive from a numerical standpoint unless some smoothing operator is applied to the kernel before differentiation. FastScat uses a high-order regulated kernel [4] that is analytic everywhere to avoid this difficulty. The third form is obtained from the second by twice integrating by parts. This reduces the singularity of the kernel to that of the Dirichlet case. It does, however, require basis functions that are differentiable.

III. HIGH-ORDER METHODS

FastScat uses patch-based basis functions for both the TM and TE polarization cases. That is to say the basis functions are nonzero only on individual patches. The patches are arbitrarily curved line segments parameterized by a function $\boldsymbol{x}(u), 0 \leq u \leq 1$. The basis functions are defined in terms of the surface parameterization according to

$$f_n(u) = \frac{\sqrt{2n+1}}{4\sqrt{g(u)}} P_n(2u - 1) \quad (17)$$

where P_n is the nth Legendre polynomial and

$$g(u) = \left(\frac{\partial x}{\partial u}\right)^2 + \left(\frac{\partial y}{\partial u}\right)^2 \quad (18)$$

is the metric for the patch [5]. The normalization factors are chosen to make the basis functions orthonormal when integrated over a patch, i.e.,

$$\int_{\text{patch}} dl\, f_m(\boldsymbol{x}) f_n(\boldsymbol{x}) = \int_0^1 du \sqrt{g(u)} f_m(u) f_n(u) = \delta_{mn}. \quad (19)$$

The contribution to the overall solution error due to surface misrepresentation can be eliminated by internally representing the surface using its exact functional form [6]. Using the combination of high-order basis functions and an exact surface representation, FastScat can obtain a high-order approximation to the smoothly varying source distribution that is to be expected on a smooth scattering surface.

By contrast, standard, low-order method of moments implementations use flat segments to approximate the surface geometry and basis functions that are constant (in the TM case) or piecewise linear (in the TE case) on a patch to approximate the sources. Representing a smoothly curved scatterer using flat segments is an example of surface representation error. Using flat segments further degrades the accuracy of the computation by introducing artificial edges, which cause spurious diffraction. Constant (or "pulse") basis functions are equivalent to the zeroth-order basis functions in FastScat; piecewise linear (or "rooftop") basis functions can be constructed from FastScat's zeroth-order and first-order basis functions. The advantage of having higher order polynomial basis functions is that they can provide accurate approximations to smooth functions more efficiently than pulse or rooftop basis functions alone can.

The third numerical method that must be high order to achieve high-order convergence in the final result involves numerical evaluation of integrals such as those in (12) and (16). Gaussian quadrature is a well-known high-order method for evaluating integrals of nonsingular integrands. The impedance matrix elements of (12) and (16) fall into this category when the regions of integration of $\boldsymbol{x}$ and $\boldsymbol{x}'$ do not intersect. Such integrals may be evaluated efficiently with Gaussian quadrature and typically are, even in standard method of moments codes. The trouble begins when the regions of integration *do* intersect, as occurs when the patches involved touch or are the same. In such cases, standard Gaussian quadrature is reduced to the status of a low-order method [7], [8]. So-called "singularity removal" (which is misnamed because, although it removes the infinity in the kernel at $\boldsymbol{x} = \boldsymbol{x}'$, it does not eliminate the singularity of the kernel at $\boldsymbol{x} = \boldsymbol{x}'$ in the strict mathematical sense) is often called upon to handle such integrals, even though it does not actually restore the high-order behavior of Gaussian quadrature.

Several schemes for high-order evaluation of singular integrands have been devised for and implemented in FastScat. One involves using quadrature rules that are specific to the singularity. For 2-D, where the singularity of the kernel is logarithmic, high-order "lin-log" rules [9] have been developed. They are designed to exactly integrate products of polynomials and logarithms. An alternate approach that is more easily extended to the 3-D scattering case, involves tampering with the kernel to eliminate the singularity at $\boldsymbol{x} = \boldsymbol{x}'$, but doing it in such a way that convolutions of the kernel with polynomial functions are still computed exactly [4]. The resulting function is regular (i.e., analytic)—hence, the name "regulated kernel". Convolutions of smooth functions with an appropriate regulated kernel may be evaluated in a high-order fashion by means of standard Gaussian quadrature. Both of these methods lead to similar results. The calculations reported in this paper were performed using a high-order regulated kernel and Gaussian quadrature.

High-order methods have the potential to greatly improve the efficiency of obtaining accurate numerical results. However, like a chain whose strength is limited by its weakest link, the convergence rate of an algorithm whose final result depends on several numerical methods, is limited by the convergence rate of its lowest order method. For scattering computations, this applies to the numerical methods used for surface representation, basis functions, and quadratures. To show how the method order of one of these components affects the rate of convergence of the full solution, it is best to vary that one while setting the method order for each of the other two components high enough that they do not contribute any noticeable error. With FastScat, the user can control the order of each of these three numerical methods.

The focus of this paper is on high-order basis functions and how they can be employed to efficiently compute accurate results. Therefore, the calculations summarized here show the effect of varying the basis function order while using exact surface representations and quadrature orders high enough that numerical integration error was negligible. In normal usage, one generally uses exact surfaces and sets the orders of the basis functions and the quadratures to be no higher than necessary to achieve the desired accuracy in the final result.

IV. RESULTS

Measuring the order of convergence of a numerical method requires observing how the error in the final result responds to changes in the discretization. For small enough discretization scales h, we expect the error to scale as $\epsilon \sim h^n$ for an nth-order numerical method.

In this next two sections, we present results of FastScat calculations on canonical 2-D geometries (a circle and an ellipse) that demonstrate how the rate of convergence varies with discretization scale size and basis function order. The third subsection is devoted to a large 2-D scattering geometry we call the "bat." The bat is prototypical of scatterers whose cross section has a large dynamic range as a function of angle. For such scatterers, the utility of a high-order scattering code becomes evident even at "practical" accuracies. Sun SPARC 10's were used for the circle calculations; the ellipse and bat calculations were performed on IBM RS/6000 computers.

A. Circle

The circle is one of the best geometries to use for investigating the convergence properties of a scattering code because it has no geometrical singularities (e.g., edges and corners) and the answer can be computed to arbitrary accuracy by summing the Mie series. This means that we can determine exactly and unambiguously what the errors are in our computed solutions, which eliminates one of the sources of disagreement about how to quantify solution accuracy.

We used FastScat to compute the bistatic cross section of 1λ-radius circles for Dirichlet and Neumann boundary conditions, corresponding to TM and TE polarizations, respectively. The circles were divided into equal segments, each segment being represented internally as a circular arc. Quadrature orders were set high enough to guarantee that numerical integrations would be accurate to better than one part in 10^{12}.

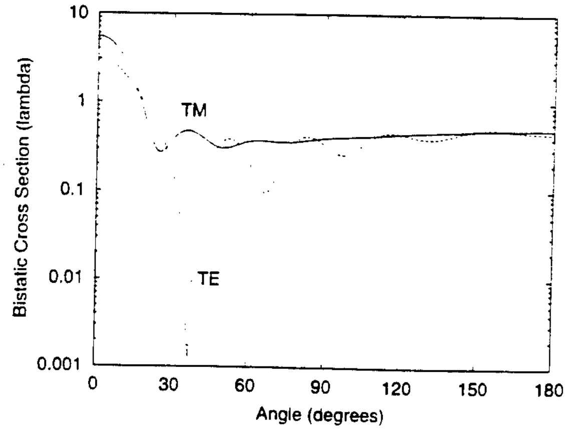


Fig. 1. Bistatic cross section of a 1λ-radius circle for TM and TE polarization (Mie series).

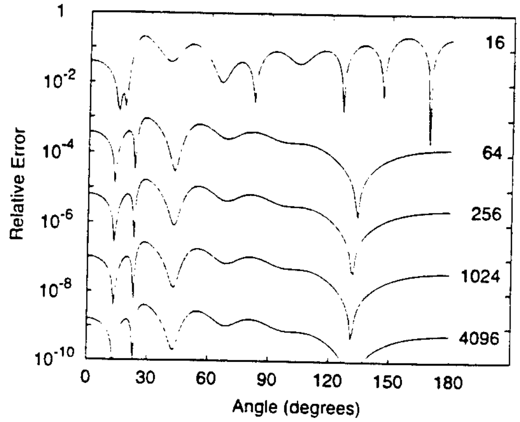


Fig. 2. Fractional difference between the cross section computed by FastScat using pulse basis functions and the exact cross section (Fig. 1) as a function of observation angle. The curves are labeled by the number of identical segments into which the 1λ-radius circle was divided.

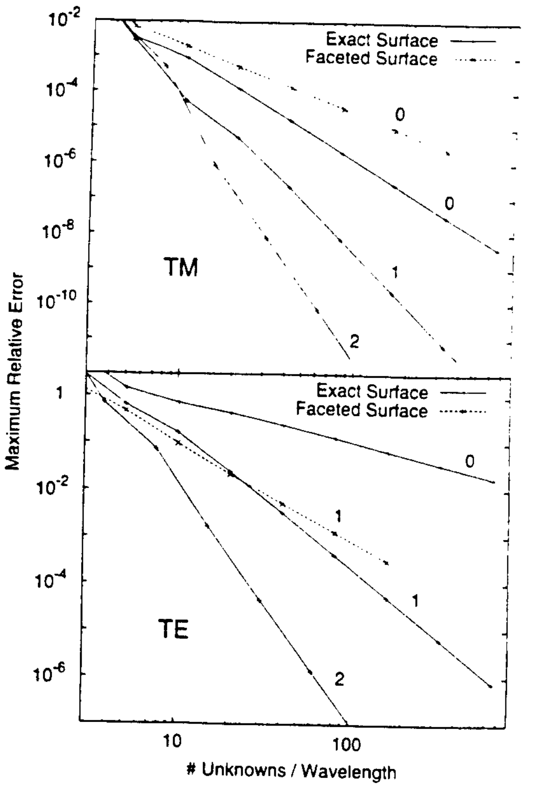


Fig. 3. Log-log plot of maximum relative error versus density of unknowns for the TM and TE polarization cases. Each set of points is labeled by basis function order.

We performed a series of calculations with different basis function orders and different numbers of segments, and compared against the exact results (Fig. 1). A sample of the results is shown in Fig. 2 for the case of zeroth-order basis functions and TM polarization. The error in the cross section varies as a function of bistatic scattering angle. It is evident, however, that, for 64 or more patches, increasing the number of patches by a factor of four reduces the overall error by a factor of about 64.

We can make a stronger quantitative statement about the discretization error if we condense the error versus angle information into a single number for each discretization. Of the many ways to do this, we have investigated three: maximum relative error, maximum error $\div$ average cross section, and root mean square (rms) error. For this particular problem, the result is essentially independent of which measure of error is chosen. Fig. 3 shows maximum relative error $(\max[|\mathrm{RCS}(\theta)/\mathrm{RCS}_{\mathrm{ref}}(\theta) - 1|])$ versus density of unknowns plotted on a log-log scale for basis function orders zero, one, and two, and numbers of patches ranging from four to 4096. Consider the TM polarization case first. The most important feature to note is that, for enough unknowns, the data fit a linear trend line whose slope increases as the basis function order increases. Since the discretization scale h is inversely proportional to the number of unknowns N, this simply reflects the fact that the error diminishes as h^m, where m increases with method order. In fact, the slopes of the lines connecting constant basis function points are close to integers—three for zeroth-order, five for first-order, and seven for second-order—indicating that the order of convergence of the cross section when using nth-order basis functions is $m = 2n + 3$.

On the same plot, we also show an example of how the surface model affects the convergence rate. The dashed curve connects points that were computed by replacing the circular arc patches with flat patches. The order of the quadratures was the same as in the previous case. For this case, however, only one basis function order is shown, namely zero. The reason is that the poor surface representation so limits the rate of convergence that increasing the order of the basis functions has essentially no effect on the accuracy of the solution. Curves for higher basis function orders are virtual copies of the zeroth-order result, shifted to higher numbers of unknowns. In all such cases, the error in the cross section is consistent with h^2 scaling.

In the TE case, the slopes of the lines connecting constant basis function points are close to one for zeroth order, three for first order, and five for second order, indicating that the order of convergence of the cross section when using nth-order basis functions is $x = 2n + 1$. The dashed curve connects points computed according to the standard method

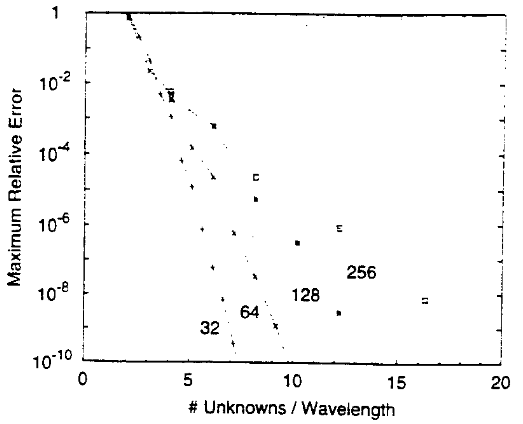


Fig. 4. Semilog plot of maximum relative error versus density of unknowns for TM scattering from a 10λ-radius circle. Points corresponding to different basis function orders for a fixed patch size are connected by lines and labeled by the number of patches.

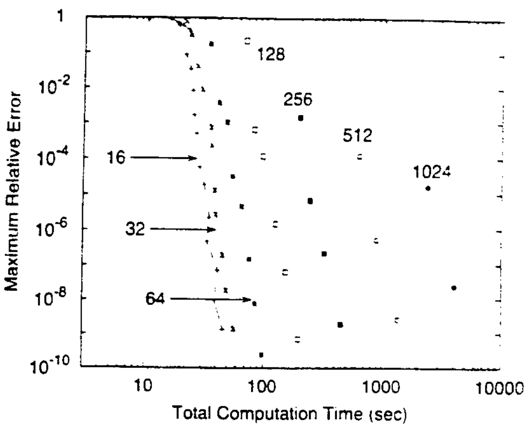


Fig. 5. Log-log plot of maximum relative error versus total computation time required to calculate the bistatic cross section of a 10λ-radius circle with TM polarization. Points corresponding to different basis function orders and a fixed patching are connected by lines, which are labeled by the number of equal arc length patches used.

of moments procedure for TE polarization, namely, by putting rooftop basis functions on a faceted approximation to the scatterer. It converges more rapidly than do the calculations that used zeroth-order (i.e., pulse) basis functions with an exact geometry representation. This is not surprising given that currents modeled by rooftop basis functions are guaranteed to be continuous across patch boundaries, whereas those modeled by pulse basis functions are not. As in the TM case, however, using higher order basis functions, whether patch-based or edge-based, does not improve the order of convergence when a low-order geometry representation is used. It only increases the number of unknowns used to achieve a given accuracy. In all such cases, the error in the cross section is consistent with h^2 scaling.

Since memory usage is proportional to N^2, these plots also show how method order affects the relationship between accuracy and memory used. For errors less than about 10^{-4} in the TM case and one in the TE case, not only are the errors in the cross sections lower when high-order methods are employed, but also the marginal cost of additional accuracy is lower.

In the plots shown so far, curves connect data points corresponding to decreasing patch sizes at a constant method order. In finite element terminology this is known as "h-refinement." As we have seen, h-refinement on a smooth scatterer results in geometric convergence in the cross section. Alternatively, one can take the same data and make a plot by connecting points of increasing method order for a fixed patch size. This is known as "p-refinement." The result of doing this for bistatic scattering from a 10λ-radius circle and TM polarization is shown in Fig. 4. The curves tend toward straight lines, which, on a semilog plot, indicates exponential convergence. Exponential convergence in the computed cross section is characteristic of p-refinement on a smooth scatterer when high-order polynomial basis functions are used.

Methods that achieve high-order convergence in general, and exponential convergence in particular, have obvious advantages for efficiently computing accurate cross sections. What may be less obvious is the fact that they facilitate accuracy estimation for computed solutions. For example, suppose we had not had an independent means (such as the Mie series for a circle) for computing a suitably accurate reference solution. We could still obtain an estimate of the accuracy of a given computed solution by comparing it to a reference solution generated by redoing the computation with an even finer discretization. To be useful, however, the reference solution must be significantly more accurate than the comparison solution. Obtaining a suitable reference solution using low-order methods may require doubling or quadrupling the number of patches, and hence the number of unknowns. The additional cost of such a calculation may be so high as to make it impractical. On the other hand, generating the reference solution by increasing the basis function order can produce a significantly better answer with only a modest increase in the number of unknowns. The increase in required memory and computation time is likewise modest. In our opinion, the widespread reliance on low order methods is what accounts for the fact that it is virtually unheard of to see accuracy estimates accompanying computed cross sections.

Another observation that may be made from Fig. 4 is that the way to achieve a high accuracy result using the least memory (i.e., fewest unknowns) is to make the patches large and put high-order basis functions on them. A look at run times instead of unknowns/memory usage leads to the same conclusion. Fig. 5 shows that for TM scattering from a 10λ-radius circle, the total computation time required to achieve a given accuracy decreases as the number of patches decreases. A point of diminishing returns is reached at around 16 patches, at which point the arc length of each patch is about 4λ. The optimum distribution of patch sizes for an arbitrary scatterer will depend on its geometry. The general rule of thumb that we follow for patching smooth scatterers is to make the patches about one wavelength long, except in regions where the geometry is strongly curved. In such regions, the patches should be some moderate fraction of the local radius of curvature.

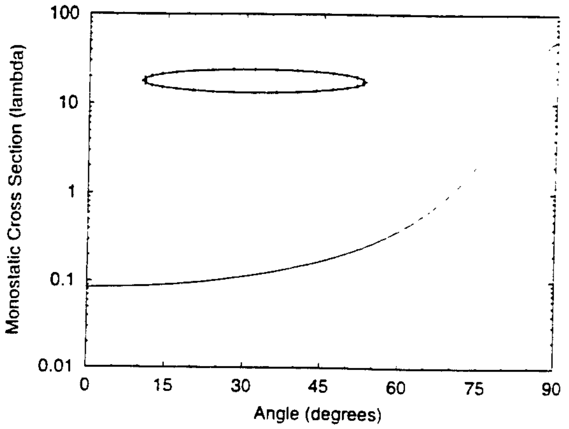


Fig. 6. Monostatic cross section of a 20λ × 2λ ellipse (shown with 32 patches) for TM polarization.

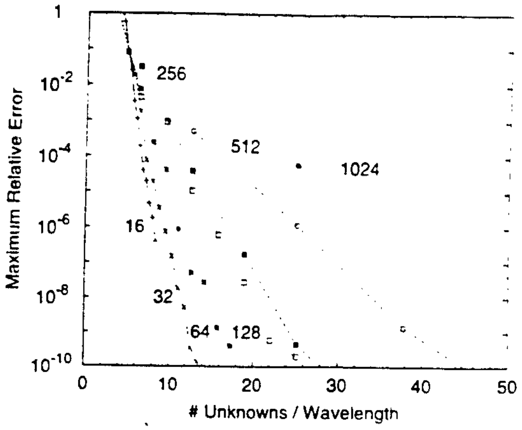


Fig. 7. Semilog plot of maximum relative error versus density of unknowns for TM scattering from a 20λ × 2λ ellipse. Points corresponding to different basis function orders for a fixed patch size are connected by lines and labeled by the number of patches.

B. Ellipse

A good candidate geometry on which to apply this rule of thumb is the 20λ × 2λ ellipse. We can describe the ellipse by the parametric equations

$$x = a\cos u \tag{20a}$$

$$y = b\sin u \tag{20b}$$

where $a = 10\lambda$ and $b = 1\lambda$. A sensible patching, which puts the highest density of patches in the most highly curved regions and vice versa for the flatter regions, is obtained if the patches cover equal increments in the parameter u, as indicated in the inset to Fig. 6.

We used FastScat to compute the monostatic cross section in TM polarization of a $20\lambda \times 2\lambda$ ellipse using several different combinations of basis function order and number of patches. In all cases, an exact surface representation was used to eliminate surface representation error, and the quadrature order was set high enough to guarantee that quadrature error would have an insignificant effect on the final accuracy. The reference solution was computed by putting tenth-order basis functions on an ellipse divided into 160 patches. Although we did not know the accuracy of the reference solution *a priori*, we have deduced from the convergence behavior of the comparison solutions that it is at least ten digits. A plot of the monostatic cross section versus angle for the reference solution is given in Fig. 6.

Fig. 7 demonstrates that one can realize exponential convergence in the cross section by using high-order basis functions with a fixed patching. In the high-accuracy regime, memory usage is optimized by using large patches and high-order basis functions. In the low-accuracy regime, the accuracy is not that sensitive to the discretization for a given density of unknowns. The accuracy at which the various curves tend to bunch up is geometry dependent, but, as a general rule, can be expected to decrease as the problem size increases.

The analog to Fig. 5 for the ellipse is Fig. 8.

C. 300λ Bat

A bat is composed of straight faces connected smoothly by circular arcs of radius R. There are two long edges of length L and six short edges, each of length $L/3$, at right angles to each other. The surfaces of the corresponding 3-D bat are assumed to be perfect conductors. It is interesting from a practical point of view because it has three high cross section specular reflection regions (one of which is the 2-D analog of a corner cube) and a low cross section everywhere else (see Fig. 9).

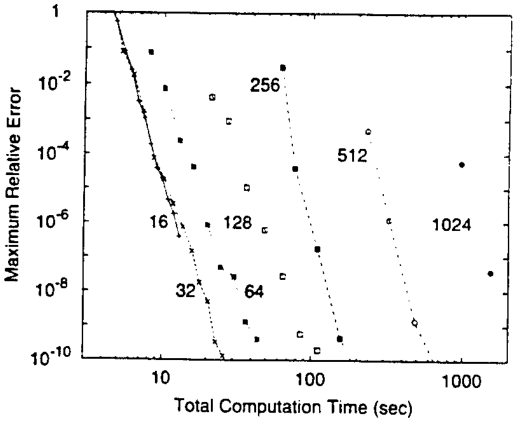


Fig. 8. Log-log plot of maximum relative error versus total computation time required to calculate the monostatic cross section of a 20λ × 2λ ellipse with TM polarization. Points corresponding to different basis function orders and a fixed patching are connected by lines, which are labeled by the number of patches used.

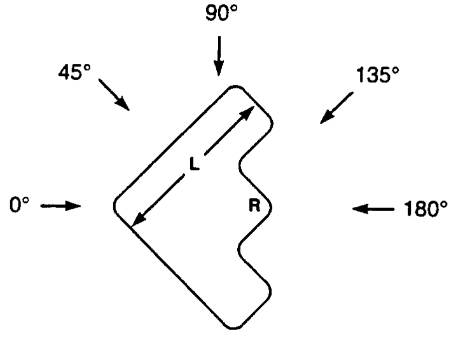


Fig. 9. "Bat" geometry.

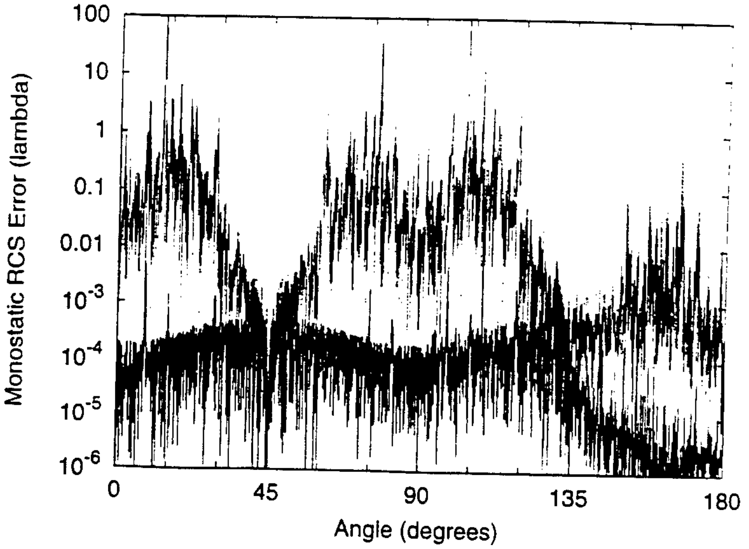


Fig. 10. Monostatic RCS of $R = 1\lambda, L = 300\lambda$ bat in TM polarization. Low-order result computed using zeroth-order basis functions on $\frac{1}{5}\lambda$ patches. High-order result computed using fourth-order basis functions on 1λ patches. Both computations used 6000 unknowns.

The results shown here are for $R = 1\lambda, L = 300\lambda$. Fig. 10 shows two computations of the monostatic cross section as a function of incidence angle for Dirichlet boundary conditions (i.e., TM polarization). One computation was performed using low-order basis functions, the other used high-order basis functions. Both calculations used an exact surface representation, quadratures good to at least eight digits of accuracy, and exactly 6000 unknowns to represent the sources. In the former case, the surface was broken up into 6000 segments, each about $\frac{1}{5}\lambda$ long, and the sources were represented by pulse basis functions (i.e., one unknown per segment). This constitutes the standard, low-order procedure (except for the exact surface representation used on the circular arcs) for solving a 2-D scattering problem with TM polarization. In the latter case, the surface was divided into 1200 patches, each about 1λ long, and basis functions up to fourth-order were employed to represent the sources (i.e., five unknowns per segment).

The two plots are very similar over a good portion of the angular range, particularly in regions of high cross section. There are narrow peaks at 45 and 135° as expected and a broader peak centered at 180°, resulting from the "corner square" effect. Note that the oscillations evident in the cross section are the result of interference, not due to any solution error. However, in the angular ranges from 0 to 30° and 60 to 120°, there are significant disagreements. The "spikes" in the upper plot Fig. 10 are suspicious looking. Which is right? How can one be sure?

Having high-order methods at one's disposal makes it possible to answer these questions with the kind of certainty that is impractical to attain with low-order methods. If we keep the same patching of the bat, but allow up to fifth-order basis functions instead, the number of unknowns increases to 7200. This corresponds to a 44% increase in the amount of memory required to store the impedance matrix and a 73% increase in the amount of CPU time required to LU decompose the impedance matrix (which is the most time-consuming step in the solution process). More importantly, allowing for one higher polynomial order to represent the sources improves the accuracy of the solution significantly. So much so that we are justified in using the fifth-order solution as a reference solution against which we can compare the lower-order solutions in order to estimate their accuracies. To compute a reference solution of comparable accuracy by the standard, low-order technique would require subdividing the 6000 patches many times into smaller patches. The number of unknowns would increase significantly. In principle, it could be done, but since CPU time for LU decomposition and memory for impedance matrix storage scale so badly with number of unknowns, the cost would be so exorbitant as to make the procedure impractical.

Fig. 11 shows plots of the differences between the fifth-order reference solution and the two solutions plotted in Fig. 10. It is evident that the fourth-order solution is the better of the two. As expected, the error is least where the cross section is highest. The estimated error of the fourth-order solution is generally below $10^{-3}\lambda$: at a few angles it rises to almost $10^{-2}\lambda$. If error bars were to be plotted on the high-order data of Fig. 10, they would all be less than the thickness of the plotted line. Fig. 11 also shows the estimated error of the low-order solution to be generally higher. Whereas it is probably acceptable over angular regions where the cross section is high, in the low cross section region the error cannot be considered acceptable, exceeding, as it does, 20 dB for certain angles. Similar results obtain for TE polarization.

V. SUMMARY

The unfavorable tradeoff between cost and problem size for method of moments solutions to scattering problems is well known and several so-called "fast" methods, such as the fast

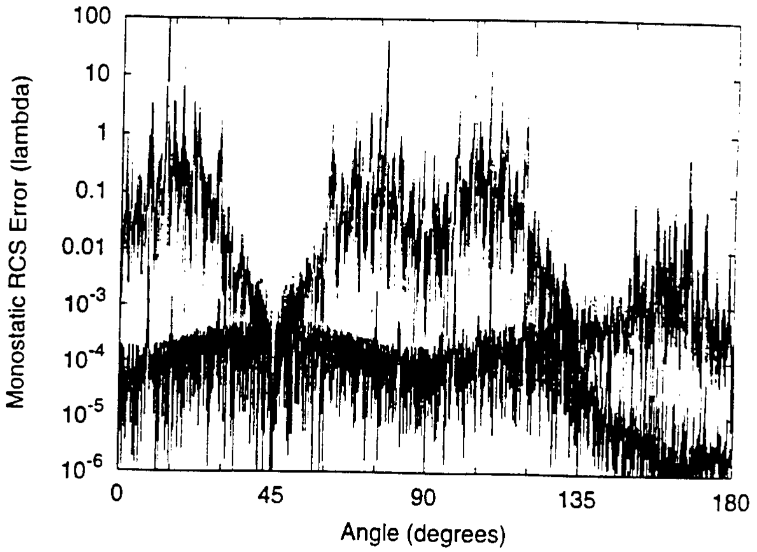


Fig. 11. RCS error with respect to reference solution computed with fifth-order basis functions on 1λ patches (7200 unknowns). Upper curve: zeroth-order calculation; lower curve: fourth-order calculation.

multipole method [10], have been devised in recent years to address it.

The subject of this paper is another tradeoff that, while no less important, is apparently much less widely appreciated. It is the tradeoff between cost and accuracy for a fixed problem size. Improving the accuracy of a computed solution requires refining the discretization, which in turn requires more memory and more computation time. With low-order methods the amount of additional computer memory and time required to achieve a more accurate result may be substantial. High-order methods are designed to make accuracy improvements much less costly.

The focus in this paper has been on using high-order basis functions to compute cross sections in 2-D. High-order basis functions are part of the triad of high-order methods that make FastScat a high-order scattering code. The results show that by using high-order methods it is possible to achieve very accurate solutions to simple scattering problems on a workstation in a reasonable amount of time. Furthermore, we have demonstrated that the solution converges at a geometric rate as a function of patch size for fixed basis function order and exponentially as a function of basis function order for fixed patch size. For high accuracies, the most computationally efficient solutions, in terms of both memory and CPU time, are produced by using high-order basis functions on large patches.

High-order methods are important for doing large problems as well. In fact, the adverse effects of a low-order discretization are likely to manifest themselves even more prominently as problems grow in size. The error caused by a low-order discretization will be particularly noticeable on scatterers whose cross section has a large dynamic range as a function of angle. We devised a large 2-D scatterer called the bat in order to demonstrate this effect. We observed that where the cross section is high, solutions computed using low-order and high-order basis functions were about the same, whereas in the more interesting regions where the cross section is low, the high-order solution is accurate while the low-order solution has significant errors. Had we used a low-order surface representation the result would likely have been worse still. The bat also demonstrated the practical utility of high-order methods for estimating the accuracy of a computed solution.

ACKNOWLEDGMENT

The authors are grateful to Prof. V. Rokhlin for many stimulating discussions regarding use of high-order methods in scattering calculations.

REFERENCES

[1] L. Hamilton, M. Stalzer, R. S. Turley, J. Visher, and S. Wandzura, "FastScat: An object-oriented program for fast scattering computation," *Sci. Programming*, vol. 2, no. 4, pp. 171–178, 1993.

[2] L. R. Hamilton, J. J. Ottusch, M. A. Stalzer, R. S. Turley, J. L. Visher, and S. M. Wandzura, "Accuracy estimation and high-order methods," in *11th Ann. Review of Progress in Applied Computational Electromagnetics*. Monterey, CA: Applied Computational Electromagnetics Society, vol. II, pp. 1177–1184, Mar. 1995.

[3] R. F. Harrington, *Field Computation by Moment Methods*. New York: Macmillan, 1968.

[4] S. M. Wandzura, "High-order regulation of singular kernels," in *Progress in Electromagnetics Research Symp.*, Univ. Washington, Seattle, WA, July 1995, p. 131.

[5] ——, "Electric current basis functions for curved surfaces," *Electromagnetics*, vol. 12, pp. 77–91, 1992.

[6] L. Hamilton, V. Rokhlin, M. Stalzer, R. S. Turley, J. Visher, and S. M. Wandzura, "The importance of accurate surface models in RCS computations," in *IEEE Antennas Propagation Soc. Symp. Dig.*, Ann Arbor, MI, vol. 3, June 1993, pp. 1136–1139.

[7] S. M. Wandzura, "Accuracy in computation of matrix elements of singular kernels," in *11th Ann. Review of Progress in Applied Computational Electromagnetics*. Monterey, CA: Applied Computational Electromagnetics Society, vol. II, pp. 1170–1176, Mar. 1995.

[8] ——, "High-order discretization of integral equations with singular kernels," in *IEEE Antennas Propagation Soc. Int. Symp. Dig.*, Newport Beach, CA, vol. 1, pp. 792–795, June 1995.

[9] J.-H. Ma. V. Rokhlin. and S. M. Wandzura. "Generalized Gaussian quadrature rules for systems of arbitrary functions." *SIAM J. Numerical Anal.*. vol. 33. pp. 971–996. June 1996.

[10] R. Coifman. V. Rokhlin. and S. M. Wandzura. "The fast multipole method: A pedestrian prescription." *IEEE Antennas Propagation Soc. Mag.*. vol. 35. pp. 7–12. June 1993.

Lisa R. Hamilton received the B.S. degree in engineering from the California State University. Northridge. in 1989.

She began her career at Hughes in 1989. Her early work included the simulation of automotive and missile control systems. and the analysis and design of infrared sensing and image processing systems. As a member of the Computational Physics Department at HRL Laboratories. she was involved for several years in the implementation of numerical methods for the calculation of electromagnetic scattering and radiation. She is currently an Advanced Project Engineer with Delphi Automotive Systems.

John J. Ottusch was born in Landstuhl. Germany. in 1955. He received the S.B. degree in physics in 1977 from the Massachusetts Institute of Technology. Cambridge. and the Ph.D. degree in physics in 1985 from the University of California. Berkeley. where he was a National Science Foundation fellow. His thesis research involved heterodyne spectroscopy of the sun in the infrared.

Since obtaining his graduate degree. he has been a member of the technical staff at Hughes Research Laboratories (now HRL Laboratories). From 1985 to 1994 his research was primarily devoted to experimental investigations of nonlinear optics. including stimulated Raman and Brillouin scattering and optical phase conjugation. Since 1994 he has been working on developing fast high-order algorithms and software for electromagnetic modeling.

Mark A. Stalzer received the B.S. degree in physics and computer science from the California State University at Northridge and the M.S. and Ph.D. degrees in computer science from the University of Southern California.

He is a Senior Research Scientist in the Computational Physics Department of HRL Laboratories. His research interests include computational electromagnetics. algorithm design. and parallel processing.

Dr. Stalzer is a member of the ACM and Sigma Xi.

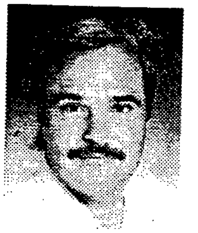
R. Steven Turley (SM'91) received the B.S. degree in physics (summa cum laude) from Brigham Young University. Provo. UT. in 1978 and the Ph.D. degree from Massachusetts Institute of Technology. Cambridge. in 1984.

In 1979 he joined Hughes Research Laboratories as a Member of the Technical Staff. He left Hughes in 1995 as a Senior Research Staff Scientist to join the faculty at Brigham Young University where he is currently an Associate Professor of Physics and Astronomy. His current areas of research interest include computational electromagnetics. XUV optics. and XUV sources.

Dr. Turley is a member of the American Physical Society. Sigma Xi. and the American Association of Physics Teachers.

John L. Visher received the B.A. degree in physics from the University of California. Santa Cruz. in 1979 and the M.A. degree in physics from Columbia University. New York. NY. in 1981.

In 1985 he joined the Hughes Reserach Laboratories. His responsibilities were in the area of HEMT and HBT testing and design. He has also designed control software for focused ion beam systems. molecular beam epitaxy chambers. and assorted other machines used for the fabrication of electronic devices. His research interests are in the area of computational electromagnetics and he has recently written a high-order integral equation code to compute the interaction of electromagnetic fields with conducting surfaces and dielectric volumes.



Stephen M. Wandzura received the B.S. degree in music from the University of California. Los Angeles. in 1971 and the Ph.D. degree in physics from Princeton University. Princeton. NJ. in 1977.

He is a Principal Research Scientist. Communications and Photonics Laboratory. HRL Laboratories. His research has been in diverse areas of scattering and propagation. His thesis research studied spin-dependent deep inelastic lepton-hadron scattering. As a National Research Fellow with the NOAA. he studied scattering of light by atmospheric turbulence and the occurrence of mountain lee waves. At HRL. he has worked on classical and quantum optics. especially the theoretical and numerical study of stimulated scattering. Since 1989. he has be studying numerical solution of scattering and radiation problems. He has published articles in *Physical Review*. *Physical Review Letters*. *Physics Letters*. *Optics Letters*. *Nuclear Physics*. *JOSA*. and other journals.

JOURNAL OF COMPUTATIONAL PHYSICS **146**, 627–663 (1998)
ARTICLE NO. CP986077

Numerical Solution of the Helmholtz Equation in 2D and 3D Using a High-Order Nyström Discretization[1]

Lawrence F. Canino, John J. Ottusch, Mark A. Stalzer, John L. Visher, and Stephen M. Wandzura

Computational Physics Department of the Communications and Photonics Laboratory, HRL Laboratories, M/S RL65, 3011 Malibu Canyon Road, Malibu, CA 90265-4799
E-mail: ottusch@hrl.com

Received December 16, 1997; revised August 11, 1998

We show how to solve time-harmonic scattering problems by means of a high-order Nyström discretization of the boundary integral equations of wave scattering in 2D and 3D. The novel aspect of our new method is its use of local corrections to the discretized kernel in the vicinity of the kernel singularity. Enhanced by local corrections, the new algorithm has the simplicity and speed advantages of the traditional Nyström method, but also enjoys the advantages of high-order convergence for controlling solution error. We explain the practical details of implementing a scattering code based on a high-order Nyström discretization and demonstrate by numerical example that a scattering code based on this algorithm can achieve high-order convergence to the correct answer. We also demonstrate its performance advantages over a high-order Galerkin code. © 1998 Academic Press

Key Words: high-order numerical method; Nyström method; boundary integral equation; Nyström discretization; local corrections; acoustic scattering; electromagnetic scattering.

I. INTRODUCTION

High-order methods are numerical methods characterized by their ability to obtain extra digits of precision with comparatively small additional effort. Scattering codes that employ high-order methods have a distinct advantage over scattering codes that use low-order methods when it comes to computing results accurately. We demonstrated this advantage with a Galerkin method of moments scattering code called FastScat™ [1, 2], which employs

---

[1] This research was supported by the Defense Advanced Research Projects Agency of the U.S. Department of Defense under Contract MDA972-95-C-0021 and by the Hughes Electronics Corporation.

0021-9991/98 $25.00
Copyright © 1998 by Academic Press
All rights of reproduction in any form reserved.

high-order methods in its geometry description, current basis functions, and quadratures. In terms of memory efficiency, the advantage of using a high-order code such as FastScat was clear. For a given number of unknowns, results obtained with FastScat were generally more accurate than those obtainable by low-order codes, with the accuracy gap widening rapidly as the number of unknowns applied to the problem was increased. In terms of CPU time efficiency, however, the advantage of using a high-order code such as FastScat was not so clear. The precomputation phase of the calculation often accounted for an undesirably large fraction of the total solution time. Although we were able to significantly accelerate the part of the precomputation phase devoted to computing near-interaction matrix elements by using high-order regulated kernels [3], the overall matrix fill procedure was still considered too slow.

The precomputation phase of a Galerkin scattering calculation is time consuming because it requires numerical evaluation of the convolution of the kernel with basis functions on every pair of source and field patches. This amounts to N^2 numerical double integrations over patches, where N is the number of unknowns. By contrast, when a point-based (Nyström) discretization is used, the impedance matrix fill step consists of nothing more than a kernel *evaluation* to fill most matrix elements and $\mathcal{O}(N)$ single integrations and some low-rank linear algebra to fill the others (specifically, the near interactions). As a result, use of a point-based discretization dramatically reduces precomputation time.

Despite its simplicity and speed advantages, the Nyström method has not been widely used for discretizing the integral equations that arise in 2D and 3D scattering problems. In fact, we know of only a few reported instances, of which [4, 5] are examples. The problem is that the conventional Nyström method [6] is designed to handle regular kernels, whereas the Helmholtz kernel for wave scattering is singular wherever the source point coincides with the field point. The standard way [6] to try to overcome this problem is to use so-called "singularity extraction," which, in practice, removes the infinity in the kernel but not the singularities in the kernel's derivatives. While singularity extraction avoids the dilemma caused by numerical evaluation of the kernel at infinities, it does not generalize easily to arbitrary surface patch geometries and it is a low-order method. In this paper, we introduce "local corrections" as a means to overcome the problems associated with kernel singularities. This enhanced Nyström discretization method has all the advantages of the standard Nyström method combined with the high-order convergence capability required to achieve error control.

This paper provides a detailed explanation for using the Nyström method to solve scattering problems in the 2D and 3D scalar cases and the 3D vector case (by which we mean electromagnetic scattering based on the Maxwell equations), as well as numerical evidence, demonstrating the method's utility. The first section reviews the traditional Nyström method for discretizing integral equations and explains how it can be adapted to handle singular kernels by incorporating local corrections. The second section discusses practical aspects of implementing a high-order Nyström code, such as appropriate surface models and meshes, choice of testing functions for computing local corrections, and how to compute scattering results. In the fourth section, we show numerical results for some 2D and 3D canonical scatterers to demonstrate that our implementation of the Nyström method achieves high-order convergence to the correct answer. We also demonstrate the run-time performance benefits of a using high-order Nyström code, compared to high- and low-order Galerkin codes, in this section. Finally, the Appendix describes how the local correction integrals for 2D scalar, 3D scalar, and 3D electromagnetic scattering can be formulated for efficient and accurate numerical evaluation.

II. NYSTRÖM METHOD

A. *Conventional Nyström Method*

The conventional Nyström method is a simple and efficient mechanism for discretization of integral equations with nonsingular kernels. Consider the integral equation

$$\phi(\mathbf{x}) = \int_S ds' G(\mathbf{x} - \mathbf{x}')\psi(\mathbf{x}') \tag{1}$$

and a quadrature rule for integrating a function $f(\mathbf{x})$ over the region S

$$\int_S ds\, f(\mathbf{x}) \cong \sum_{n=1}^{N} \omega_n f(\mathbf{x}_n). \tag{2}$$

Such a quadrature rule will be provided by Gauss–Legendre or Gauss–Jacobi rules on a parameterization of S, so that the weights ω_n will be the products of the elementary weights w_n with the Jacobian of the parameterization:

$$\omega_n = \sqrt{g(u_n)}w_n, \tag{3}$$

$$\mathbf{x}_n = \mathbf{x}(u_n), \tag{4}$$

where u_n are the abscissae of the elementary rule, $\mathbf{x}(u)$ is the mapping function of the surface S, and $g(u)$ is the determinant of the mapping metric. The extension to patched parameterizations is straightforward.

The Nyström discretization of a function on S is simply the tabulation of the function at the quadrature points $\mathbf{x}_n$:

$$\psi_n = \psi(\mathbf{x}_n). \tag{5}$$

To discretize integral Eq. (1), we simply form a matrix from the kernel:

$$\phi_m = \sum_{n=1}^{N} \omega_n G(\mathbf{x}_m - \mathbf{x}_n)\psi_n. \tag{6}$$

This discretization has an error of the same order as the underlying quadrature rule [7]. In other words, if the surface S is smooth, ϕ and $G(\mathbf{x} - \mathbf{x}')$ are regular functions, and if a high-order quadrature rule is used, then the solution to Eq. (6) represents a high-order approximation to the exact solution.

Unfortunately, the kernels $G(\mathbf{x} - \mathbf{x}')$ for wave scattering are not regular. Instead, they have singularities (or even hypersingularities) at short distances. With such kernels it is often not even possible to make a matrix out of the kernel because its value is undefined when $\mathbf{x} = \mathbf{x}'$. Even if the kernel were finite at vanishing separation, a kernel singular in its higher derivatives would spoil the high-order properties of the above prescription.

B. *High-Order Nyström Method for Singular Kernels*

We have adapted the Nyström method to handle singular kernels, without sacrificing high-order convergence, by incorporating Strain's method [8] for obtaining high-order quadrature rules for singular functions. The essence of the method is that by computing convolutions of the kernel with a suitable set of testing functions, it is possible to determine how to adjust the quadrature rule so that it is just as accurate near the singularity as far from it. The beauty of the method is that these quadrature rule modifications are required only in the vicinity of the singularity, hence the name *local corrections*.

Conceptually, local corrections may be viewed as adjustments to the quadrature weights (at the original set of sample points) that are required to make the quadrature rule high-order accurate when the (singular) function $G(\mathbf{x}-\mathbf{x}')$ is included in the integrand. In practice, since quadrature weights and discretized kernel terms always enter into the quadrature rule as product pairs, one can equally well "locally correct" the discretized representation of kernel and keep the original quadrature weights. This is the preferred approach because the modified representation of the kernel has no infinities. We can write the "corrected" matrix representation of the kernel as

$$\tilde{G}_{mn} \equiv \begin{cases} L_{mn}, & \text{when } \mathbf{x}_n \in D_m, \\ G(\mathbf{x}_m - \mathbf{x}_n), & \text{otherwise,} \end{cases} \tag{7}$$

where L_{mn} is a (sparse) matrix of local corrections whose entries are nonzero only for source points $\mathbf{x}_n$ within a small domain D_m centered on the field point $\mathbf{x}_m$. For $|\mathbf{x}_m - \mathbf{x}'|$ sufficiently large (i.e., outside the local correction domain D_m), $G(\mathbf{x}_m - \mathbf{x}')$ is a smoothly varying function of position and the underlying quadrature rule provides a high-order approximation to the desired integral. Close to the singularity, on the other hand, the singular nature of the kernel spoils the high-order behavior of the underlying quadrature rule, and it becomes necessary to use locally corrected values for the kernel instead of $G(\mathbf{x}_m - \mathbf{x}_n)$ in order to achieve high-order convergence. The mechanism for computing the local corrections for a given set of source points is explained below. The size of the local correction domain is discussed in Section III.D.

The underlying quadrature rule is exact for integration of a certain class of functions (typically polynomials). We choose the local corrections to make convolution of the singular kernel with the same class of functions exact. They are obtained by solving the linear system

$$\sum_n \omega_n L_{mn} f^{(k)}(\mathbf{x}_m - \mathbf{x}_n) = \int_{D_m} ds' G(\mathbf{x}_m - \mathbf{x}') f^{(k)}(\mathbf{x}_m - \mathbf{x}'), \tag{8}$$

which represents K constraints (one for each testing function $f^{(k)}$) on J local correction coefficients (one for each of J source points in the vicinity of the mth field point). The integral over D_m can be obtained by oversampling the region of integration until the result has converged to the desired accuracy. The nonzero components of the mth row of the local correction matrix are obtained by inverting the (small) system of equations above, either by factorization (via LU decomposition) if $J = K$ or by singular value decomposition (SVD) if $J \neq K$. Computing local corrections is the most time consuming step of the precomputation phase. Fortunately, it needs to be done only once at every sample point.

C. *High-Order Nyström Method Advantages*

There are several reasons for using the Nyström method to achieve a high-order discretization:

- *Faster precomputation.* Unlike the Galerkin method, which requires N^2 numerical double integrations to fill the impedance matrix, the Nyström method requires less than N^2 kernel evaluations and $\mathcal{O}(N)$ calculations of local correction coefficients (each of which involves a small number of adaptive integrations and a low-rank matrix inversion). An additional acceleration is possible when multiple solutions are desired at different frequencies. This comes about because a frequency-dependent Helmholtz kernel can be written as the product of a smoothly varying, frequency-dependent function and a frequency-independent Laplace kernel. Once the local corrections for the Laplace kernel have been computed, they can be used with minor modification at any frequency.
- *Elimination of multipatch, parametric basis functions.* Conventional method of moments scattering codes require basis functions with a certain level of continuity (in the surface parameterization) across patch boundaries to facilitate differentiation. For example, an important property of the popular RWG [9] basis functions for electromagnetic scattering is that their normal components are continuous across patch boundaries. One can also use high-order extensions to the RWG basis functions [10], although we have found that implementing these basis functions in a scattering code can be both complicated and inconvenient, especially for arbitrary, curved surfaces. Fortunately, for high-order codes the requirement to use elemental sources with guaranteed continuity between patches disappears because continuity of the source distribution is achieved as a natural consequence of accurately solving the integral equation. (The reason this is so has to do with the fact that the error caused by not enforcing continuity of the elemental sources is comparable to the error of the underlying discretization. With a low-order discretization (e.g., RWG basis functions on flat patches), continuity enforcement has a significant payoff because the error in the underlying discretization is also significant. With a high-order discretization, where the error due to the underlying discretization can more easily be made insignificant, the situation is reversed. Thus, for high-order codes, whether Galerkin or Nyström, the benefits of enforcing source continuity between patches do not outweigh the inconveniences.)
- *More amenable to fast solution algorithms.* Implementation of a fast method that requires segregation of the discretized scatterer into groups (such as the fast multipole method (FMM) [11] or adaptive integral method (AIM) [12]) is simpler and more natural with a point-based discretization. When a Galerkin implementation with overlapping basis function domains is employed, the fast algorithm is either more complicated (because multipatch basis functions must be split apart) or less efficient (because the groups are larger). A Galerkin implementation that uses high-order basis functions (even those confined to single patches) cannot achieve optimum efficiency from the FMM because high-order basis functions are used to their greatest advantage on patches larger than a wavelength, whereas optimum use of the FMM favors groups smaller than a wavelength. In a Nyström discretization, the groups consist of individual sample points on the surface, so no such grouping restrictions apply.
- *Iterative solver memory reduction.* With the Nyström method, the memory requirement for an iterative solver using the full impedance matrix can be reduced from $\mathcal{O}(N^2)$ (storing the full impedance matrix) to $\mathcal{O}(N)$ (storing only the sparse local correction matrix). This is practical because reconstruction of the unsaved portions of the impedance matrix only requires evaluations of the kernel, which are fast. If the FMM is used to represent the far interactions, the storage requirement goes from $\mathcal{O}(N^{5/4})$ in the single-stage case [13] to $\mathcal{O}(N \log(N))$ in the multilevel case [14].

- *Symmetry exploitation.* When basis functions are used, it is more complicated to reflect geometrical symmetries in the matrix representation. It may be necessary to explicitly consider basis function transformation properties and to provide special treatment for some variables (e.g., the coefficients of basis functions whose domains intersect reflection planes). In the Nyström case, the representation of symmetries is much simpler.

III. PRACTICAL CONSIDERATIONS

A. *Surface Description*

Without a high-order surface description, a high-order Nyström discretization is of little benefit. For example, representing a curved surface by means of flat facets limits the rate of solution convergence to low order whether or not the rest of the discretization method is high order. Ideally, the internal representation of the surface exactly matches the physical surface. Such a representation is possible for idealized curved shapes such as circles, ellipses, ogives, etc. in 2D, and spheres, ellipsoids, etc. in 3D. For curved objects of more practical interest, a high-order description of the physical surface may be given by high-order parametric representations such as bicubic splines or NURBS (nonuniform rational B-splines). As these are often the representations used by a CAD program to describe the object as it is being designed and built, it is appropriate that we should also use them for electromagnetic or acoustic modelling purposes.

Use of a high-order surface description is distinguished from that of a faceted description in that the subdivision of the surface into patches is typically done once and refining the discretization to improve accuracy is accomplished by increasing the order of the quadrature rule (which increases the number of sample points per patch).

B. *Meshing*

The essence of a point-based discretization is the tabulation of functions at a set of points lying on the surface. This need not have anything to do with subdividing a surface into patches. Indeed, in the 2D case, patches can be done away with entirely on closed surfaces (i.e., closed curves) parameterized by arc length, because the trapezoidal rule is a high-order quadrature rule for periodic functions. In 3D, however, global parameterizations with natural, high-order quadrature rules are much harder to come by, so subdivision of a surface into patches, each of which comes with its own high-order quadrature rule, becomes a practical necessity.

Since patches are introduced solely for the purpose of providing ready-made, high-order quadrature rules on the surface, the job of meshing a surface is simpler and less restrictive. Specifically, whereas a mesh designed for use with RWG-type basis functions is not allowed to have a vertex in the middle of an edge, there is no such restriction on a mesh designed for a point-based discretization. The only practical restrictions are that the mesh cover the surface and that the patches not be so distorted or curved that the supposedly high-order quadrature rules are not actually high order.

C. *Testing Functions*

The choice of testing functions goes together with the choice of quadrature rule. If the quadrature rule is designed to efficiently integrate regular functions, the testing functions should be regular functions of increasing order. In locations where singular behavior of the source function is expected, such as near geometric singularities (e.g., edges and corners), it may be desirable to apply a different quadrature rule and use appropriately singular testing functions [15]. For purposes of this discussion, we will assume the scattering surface and the sources are smooth functions of position. Any departures from regularity can be accommodated reasonably efficiently by tapering the size of the patches in the direction of the singularity.

Testing functions may be global or local. Examples of global testing functions are monomials in the surface parameter u in the 2D case, and powers of x, y, and z in the 3D case. The advantage of using global testing functions to compute local corrections on smooth surfaces is that such testing functions are manifestly continuous across patch boundaries, just like the sources. Sometimes enforcing continuity is a mistake, however, such as when the field point and source patch are near each other but on separate, unconnected surfaces. Global testing functions can also perform badly near geometric singularities such as a right-angle bend. Local testing functions (i.e., testing functions confined to individual patches) do not take full advantage of the guaranteed continuity of the sources on touching patches but are the preferred choice because they are simpler to implement and more robust.

With local testing functions, the local corrections for a given field point can be computed on a patch by patch basis. Thus, the number of points whose quadrature weights are being corrected always equals the number of sample points on the patch. Doing this has the side benefit of keeping down the size of the local correction linear systems that must be solved when it becomes necessary to compute local corrections for points on several patches.

The number of local testing functions to use is still a free parameter. In 2D, where use of a Gauss–Legendre rule of order M allows exact integration of polynomials up to order $2M$ (i.e., degree $2M - 1$), it makes sense to use as many testing functions as there are points to locally correct. In effect, the singular kernel and the unknown source function are both being approximated to order M, which means the order of approximation for the product is $2M$. This results in an exactly determined system of equations for computing local corrections.

In 3D, if a Gauss–Legendre product rule of order $M_x M_y$ is used on quadrilateral patches, the natural number of local testing functions to use is $4M_x M_y$. This leads to an exactly determined system. If the patches are triangles, one can use the quadrature rules of Lyness and Jespersen [16] and their higher-order extensions. For these triangle rules, a natural correspondence between the number of sample points and the maximum testing function degree is less obvious. When the number of sample points and the number of testing functions are not the same, they can at least be made close, in which case the nonsquare linear system of equations for the local corrections can be solved by computing a pseudoinverse using SVD. In our experience, local correction systems that are square or nearly square perform best.

C.1. Two-dimensional scalar testing functions. Monomials of increasing degree in the parameterization, i.e., $f^{(k)}(u) = u^k$, are the simplest testing functions, but they can also be troublesome when using high-order rules because they produce linear systems for computing local corrections whose condition number grows exponentially with degree. The alternative we favor is orthogonal polynomials such as Legendre or Lagrange polynomials. With either of these polynomials as testing functions, it takes a little longer to compute the integral on the right-hand side of Eq. (8), but the linear system is well conditioned for all polynomial degrees. In addition, if the number of testing functions K equals the number of source points whose quadrature weights are being corrected J, then the system is orthogonal and the matrix consisting of the K testing functions evaluated at the J different source points can be inverted simply by transposition.

C.2. Three-dimensional scalar testing functions. The trade-off between the simplicity of monomials and the better conditioning behavior associated with orthogonal polynomials exists also in the 3D cases. In 3D, however, our experience have been confined to testing functions of a low enough degree that use of monomial functions generally does not pose any serious trouble. On triangular patches, we use testing functions of the form

$$f^{(k)}(\mathbf{u}) = (u^1)^m (u^2)^n, \tag{9}$$

where u^1 and u^2 are the parameters of the surface description and the exponents obey $0 \le m, n \le M$ and $0 \le m + n \le M$ for some maximum testing function degree M.

C.3. Three-dimensional vector testing functions. In this case, vector testing functions locally tangent to the surface are required; continuity of the testing functions between adjacent patches is not. A natural set of basis vectors is given by the derivatives of the surface with respect to the two surface parameters u^1 and u^2. We use testing functions of the form

$$\mathbf{t}_\nu^{(k)}(\mathbf{u}) = \frac{\partial_\nu \mathbf{x}(\mathbf{u})}{\sqrt{g(\mathbf{u})}} f^{(k)}(\mathbf{u}), \tag{10}$$

where $\nu = 1, 2$ and the scalar functions $f^{(k)}(\mathbf{u})$ are the same as those used in the 3D scalar case. This form for the testing functions has the property that the surface divergence of $\mathbf{t}_\nu^{(k)}$ is

$$\nabla \cdot \mathbf{t}_\nu^{(k)}(\mathbf{u}) = \frac{\partial_\nu \mathbf{x}(\mathbf{u})}{\sqrt{g(\mathbf{u})}} \cdot \left(\nabla f^{(k)}(\mathbf{u})\right) = \frac{\partial_\nu \mathbf{x}(\mathbf{u})}{\sqrt{g(\mathbf{u})}} \cdot \left(\sum_{\alpha\beta} g^{\alpha\beta} \partial_\alpha f^{(k)}(\mathbf{u})\, \partial_\beta \mathbf{x}\right) \tag{11}$$

since $\partial_\nu \mathbf{x}(\mathbf{u})/\sqrt{g(\mathbf{u})}$ is divergenceless (see Appendix C). This form for the divergence of $\mathbf{t}_\nu^{(k)}(\mathbf{u})$ (which enters into the computation of local corrections for the hypersingular kernel) has the especially desirable property that it avoids the need to compute second or higher order derivatives of the surface.

D. *Extent of Local Correction Domain*

When local testing functions are used, the region over which local corrections should be computed always includes the patch containing the field point, and it extends out to include other patches until the underlying quadrature rule is accurate enough to replicate the exact answer to within a desired tolerance. Since the testing functions have local support, the problem of computing local corrections for a region containing several patches decouples naturally into several smaller local correction problems, one for each patch. The tolerance should be based on an estimate of the optimum accuracy that the particular discretization could achieve; there is, after all, little to be gained by trying to evaluate the impedance matrix more accurately than what is warranted by the discretization. The integrals on the right-hand side of Eq. (8) can be computed by adaptive integration to comparable accuracy.

E. *Local Corrections for "Regular" Parts of the Kernel*

In principle, it is unnecessary to compute local corrections for regular components of the kernel because they will be efficiently integrated by a quadrature rule of sufficiently high order. If such components are strongly peaked, however, the required order may be so high that it is computationally more efficient to treat them as if they were singular and compute local corrections for them. For example, the scalar kernel $\hat{\mathbf{n}}' \cdot \nabla' G(\mathbf{x}, \mathbf{x}')$ in 2D or 3D is a strongly peaked function of $\mathbf{x}'$ when the field point $\mathbf{x}$ is close to, but not on, the source patch. This situation arises in the analysis of scattering from thin layers, for example. One way to handle this problem is to put a fine discretization on each layer, in effect subdividing the strongly peaked kernel function into small parts, each of which is relatively smooth. This procedure is inefficient, however, because it uses many more sample points than are warranted by the expected spatial structure of the source. A better approach would be to discretize each layer densely enough to adequately represent the sources and compute local corrections for the strongly peaked kernel. Computing such local corrections can be a nontrivial task by itself, but one might expect that the extra time spent in precomputation would be compensated by a less time-consuming solution phase.

F. *Using the Results*

F.1. Computing scattered fields. The amplitude of a scattered wave can be computed by convolving the scattered wave with the source distribution. Even though a Nyström discretization specifies the source only at a finite set of points, these points are ideally suited for evaluating integrals in a high-order fashion by virtue of Eq. (2). For example, the amplitude $F(\mathbf{k})$ for 3D scalar scattering of a source distribution $\psi(\mathbf{x})$ on a surface S with Neumann boundary conditions (i.e., $\hat{\mathbf{n}} \cdot \nabla\psi(\mathbf{x}) = 0$ for $\mathbf{x}$ on S) into the plane wave given by $\phi(\mathbf{x}) = e^{i\mathbf{k}\cdot\mathbf{x}}$ is

$$F(\mathbf{k}) = \frac{1}{4\pi} \oint_S ds(\hat{\mathbf{n}} \cdot \nabla\phi^*(\mathbf{x}))\, \psi(\mathbf{x}) \tag{12}$$

$$\cong \frac{1}{4\pi} \sum_i \omega_i (\hat{\mathbf{n}}(\mathbf{x}_i) \cdot \nabla\phi^*(\mathbf{x}_i))\, \psi(\mathbf{x}_i), \tag{13}$$

where the sum is over all quadrature points and $*$ indicates complex conjugation. The extensions to other forms of scattering, whether near- or far-field, are straightforward.

F.2. Source interpolation. When a scattering problem is solved using a Galerkin scattering code, it is obvious how to compute the value of the source distribution at any point on the surface because the solved-for coefficients multiply basis functions that are uniquely defined at every point on the surface. The Nyström discretization, on the other hand, returns values of the sources only at a finite set of discrete sample points, so that determining the value of the source distribution at a point that is not part of this set requires interpolation.

When the scattering computation is performed using a second kind integral formulation, one can use the original Nyström interpolation formula, augmented by local corrections, to interpolate the source distribution. As an example, if the magnetic field integral equation (MFIE) is used to solve for the electric current distribution $\mathbf{J}(\mathbf{x})$ induced on a perfectly electrically conducting (PEC) scatterer by an incident magnetic field $\mathbf{H}^{\text{inc}}(\mathbf{x})$, one can write the current at any point $\mathbf{x}$ on the surface S as [17]

$$\mathbf{J}(\mathbf{x}) = 2\hat{\mathbf{n}}(\mathbf{x}) \times \left[\mathbf{H}^{inc}(\mathbf{x}) - \oint_S ds'\, \nabla' G(\mathbf{x}, \mathbf{x}') \times \mathbf{J}(\mathbf{x}')\right]. \tag{14}$$

We obtain an interpolation formula from this continuous equation by using Eq. (2) to approximate the integral, i.e.,

$$\mathbf{J}(\mathbf{x}) = 2\hat{\mathbf{n}}(\mathbf{x}) \times \left[\mathbf{H}^{inc}(\mathbf{x}) - \sum_i \omega_i \nabla' G(\mathbf{x}, \mathbf{x}_i) \times \mathbf{J}(\mathbf{x}_i)\right]. \tag{15}$$

where the sum over i extends over all sample points on S. Of course, to make this a high-order interpolation formula, it may be necessary to compute local corrections to the quadrature rule at source points in the vicinity of the field point $\mathbf{x}$.

Another interpolating function, which does not require computing new local corrections and is usable with first or second kind integral formulations, takes the form of a linear combination of the functions that are integrated exactly by the underlying quadrature rule. The coefficients may be determined by convolving the source with the projection operator

$$I(\mathbf{x}, \mathbf{x}') = \sum_{m,n} f_m(\mathbf{x})(N^{-1})_{mn} f_n(\mathbf{x}'), \tag{16}$$

where the summation extends over all functions $f_i(\mathbf{x})$ for which the quadrature rule is exact, and N is a normalization matrix whose components are given by

$$N_{mn} = \int_S ds\, f_m(\mathbf{x}) f_n(\mathbf{x}). \tag{17}$$

If the $f_i(\mathbf{x})$'s are orthonormal over S, then N is simply the identity matrix. Convolution with $I(\mathbf{x}, \mathbf{x}')$ eliminates the part of a function that is orthogonal to all the $f_i(\mathbf{x})$'s. If we evaluate the convolution of $I(\mathbf{x}, \mathbf{x}')$ with the source function by means of the underlying quadrature rule, we arrive at the following source interpolation function $s(\mathbf{x})$, which only requires knowledge of the source at the discrete set of sample points $s(\mathbf{x}_i)$:

$$s(\mathbf{x}) = \sum_{m,n} f_m(\mathbf{x})(N^{-1})_{mn} \sum_i \omega_i f_n(\mathbf{x}_i) s(\mathbf{x}_i). \tag{18}$$

The summation over i in the above equation extends over all sample points.

IV. RESULTS

This section is composed of two parts. The objective of the first part is to show that our most recent version of FastScat, which uses a Nyström discretization, achieves high-order convergence to the correct answer for a few small, benchmark problems from 2D scalar and 3D vector scattering. In the second part, we benchmark the performance of this code against two Galerkin codes, comparing them on the basis of CPU time and solution accuracy.

A. *Validation*

The most common practice seen in the literature for demonstrating the validity of a scattering code is to show that the results obtained from the code with a *particular* discretization compare favorably to a reference solution obtained from a series solution, another scattering code, or measurements. Individual results such as this, while useful and necessary, say nothing about the convergence properties of the algorithm on which the code is based. To show how an algorithm converges, one must compute results with a sequence of increasingly fine discretizations and observe whether and how the results converge to the correct answer.

This is especially important when validating a (purportedly) high-order code. One cannot expect to enjoy the benefits of a high-order code (more accurate solutions, solution error control, etc.) on large scattering problems without first verifying that the code achieves high-order convergence on small scattering problems (where it is easier to generate solutions with very small errors). The order of convergence of a numerical method relates to the rate at which the error in the computed solution decreases as the discretization scale decreases. For small enough discretization scales h, the error in the solution computed by a pth-order method scales as h^p. The results presented in this section will be shown to follow this scaling law.

The benchmark problems include a circle and an ellipse in 2D, and a sphere and an ellipsoid in 3D. In the 2D scalar scattering cases, results for both Dirichlet and Neumann boundary conditions on the surface will be presented; in the 3D vector (electromagnetic) scattering cases, it will be assumed that the surfaces are perfect conductors. The surface boundary conditions are chosen mainly for simplicity; similar convergence behavior has been shown for other types of boundary conditions (such as impedance boundary conditions and dielectric interfaces) as well.

A.1. Two-dimensional scalar. We solved four different integral equations to obtain 2D scalar scattering results. For Dirichlet boundary conditions (which correspond to the TM polarization case of electromagnetic scattering from an object with cylindrical symmetry) the first-kind integral equation is

$$\phi^{\text{inc}}(\mathbf{x}) = -\oint_C dl'\, G(\mathbf{x}, \mathbf{x}')\,\sigma(\mathbf{x}'), \tag{19}$$

and the second-kind equation is

$$-\hat{\mathbf{n}} \cdot \nabla\phi^{\text{inc}}(\mathbf{x}) = \frac{1}{2}\sigma(\mathbf{x}) + \oint_C dl'(\hat{\mathbf{n}}' \cdot \nabla' G(\mathbf{x}, \mathbf{x}'))\,\sigma(\mathbf{x}'). \tag{20}$$

In these equations $\phi^{\text{inc}}(\mathbf{x})$ is the incident scalar field, $G(\mathbf{x}, \mathbf{x}')$ is the 2D scalar kernel, and $\hat{\mathbf{n}}$ and $\hat{\mathbf{n}}'$ are the unit normals to the contour C at the field and source points, respectively. For this polarization case, the 2D scalar source σ is proportional to the z component of the electric current $\mathbf{J}$ in the corresponding 3D vector problem, assuming $\mathbf{z}$ is the axis of translational symmetry.

For Neumann boundary conditions (which correspond to the TE polarization case of electromagnetic scattering) the first-kind integral equation is

$$\hat{\mathbf{n}} \cdot \nabla\phi^{\text{inc}}(\mathbf{x}) = \oint_C dl'(\hat{\mathbf{n}} \cdot \nabla)(\hat{\mathbf{n}}' \cdot \nabla' G(\mathbf{x}, \mathbf{x}'))\,\psi(\mathbf{x}') \tag{21}$$

and the second-kind equation is

$$\phi^{\mathrm{inc}}(\mathbf{x}) = \frac{1}{2}\psi(\mathbf{x}) - \oint_C dl'(\hat{\mathbf{n}}' \cdot \nabla' G(\mathbf{x}, \mathbf{x}'))\, \psi(\mathbf{x}'). \tag{22}$$

For this polarization case, the electric current $\mathbf{J}$ in the corresponding 3D vector problem, assuming $\mathbf{z}$ is the axis of translational symmetry, is related to the 2D scalar source ψ by

$$\mathbf{J} = \psi \hat{\mathbf{n}} \times \hat{\mathbf{z}}. \tag{23}$$

A combined field equation can be obtained in either case by adding the first and second kind equations together using an appropriate combination coefficient [18]. Although no combined field equation results are reported here, it should be noted that use of a combined field formulation is often recommended because, by being insensitive to internal resonances, it can improve the condition number of the impedance matrix.

A.1.a. *1 λ-radius circle.* A circle is the ideal problem for benchmarking a high-order scattering code because its surface is smooth and easy to define exactly, and its cross section can be determined, for purposes of comparison, to arbitrary accuracy using the Mie series [19]. We used FastScat to compute the bistatic cross section of a 1 λ-radius circle whose surface obeys either Dirichlet or Neumann boundary conditions, which correspond to TM and TE polarizations, respectively. Meshing the circle consisted of dividing it into circular segments of equal arc length. Nyström sample points were distributed on each patch (parameterized by arc length) according to a Gauss–Legendre integration rule of a given order and Legendre polynomial testing functions up to half this order were used for computing local corrections. The resultant local correction linear systems are square.

We performed a series of calculations with different discretizations (i.e., different numbers of patches and different Nyström quadrature orders) and compared the results to the Mie series results (shown in Fig. 1). For a given Nyström quadrature order (which we henceforth

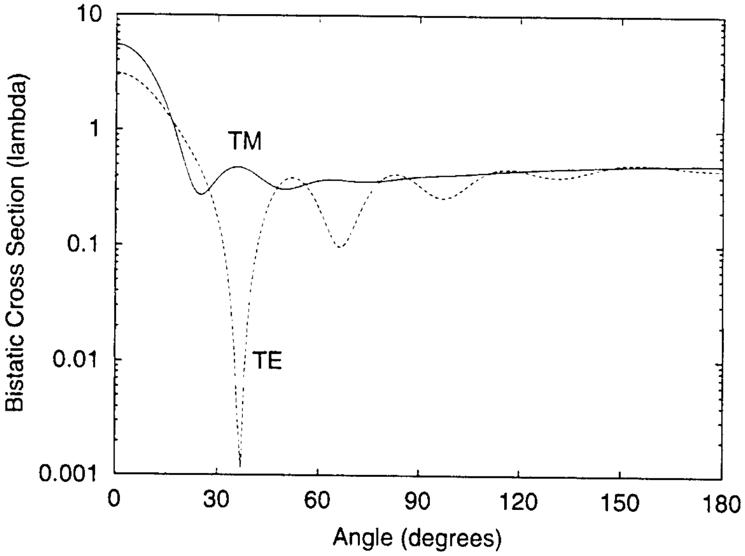


FIG. 1. Bistatic cross section of a 1 λ-radius circle for TM and TE polarizations computed by the Mie series.

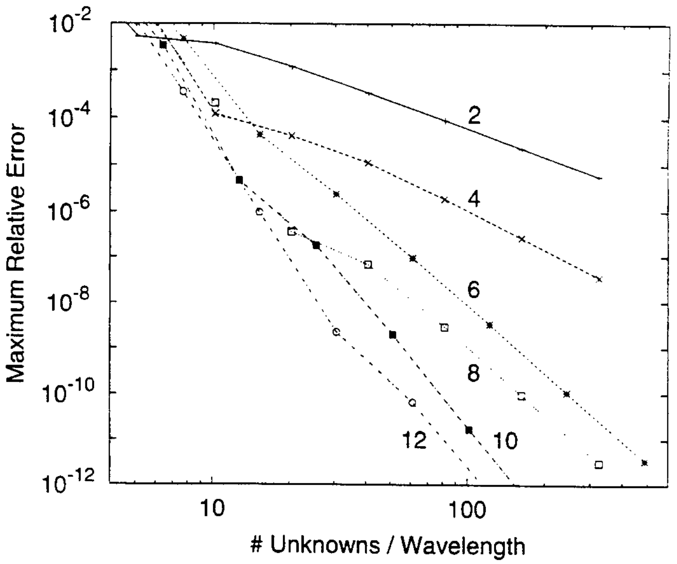


FIG. 2. Log–log plot of maximum relative error vs unknown density for 1 λ-radius circle and TM polarization. Each set of points is labeled by Nyström order.

abbreviate to Nyström order), as the size of the patches decreases, the difference between the exact result and the FastScat calculation also decreases.

A more quantitative measure of convergence behavior is given in Fig. 2, where we have plotted maximum relative error (defined as $\max[|\sigma(\theta)/\sigma_{\text{ref}}(\theta) - 1|]$, where $\sigma(\theta)$ and $\sigma_{\text{ref}}(\theta)$ are the calculated and exact cross sections, respectively, for $\theta = 0$ to 180° in 1° increments) versus the density of unknowns for a first-kind integral formulation of the TM polarization case. The number of patches spanning the circle ranged from 4 to 2048 and the Nyström order ranged from 2 to 12. One of the important features to note is that, with enough unknowns, the data fit a linear trend line whose slope increases as the Nyström order increases. Since the discretization scale h is inversely proportional to the density of unknowns, a linear fit on a log–log plot of error versus unknown density reflects the fact that the error scales asymptotically as h^p, where p (the order of convergence) increases with Nyström order. Large values of p signify a high-order algorithm. For the lower Nyström orders, the slopes of the lines connecting points of a given order are observed to be close to integers, namely 2 for order 2; 3 for order 4; and 5 for orders 6 and 8. The slopes for orders 10 and 12 are still higher, although even at the highest sampling densities used, the discretization error has not yet reached the asymptotic regime where each would be expected to have a slope of 7.

The results for the second-kind integral formulation of the TM polarization case are very similar. This should not be too surprising, since, despite the additional derivative, the singularity of the kernel is no worse than $\log(r)$.

The corresponding plot for the TE polarization case, also using a first-kind integral formulation, is shown in Fig. 3. In the TE case, however, the first-kind integral equation involves the 2D hypersingular kernel. The effect of using a more singular kernel is that the source must be represented more accurately in order to achieve the same accuracy in the cross section, or equivalently, that an equally well represented source (i.e., one employing the same collection of unknowns) produces a less accurate value for the cross section. This

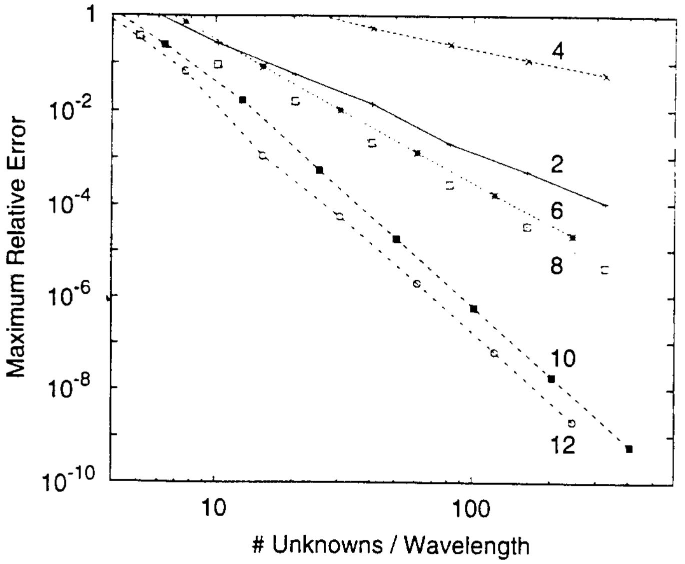


FIG. 3. Log–log plot of maximum relative error vs unknown density for 1 λ-radius circle and TE polarization. Each set of points is labeled by Nyström order.

is easily seen by comparing Figs. 2 and 3. For a given discretization, the calculated cross section for the TE case is two or more orders of magnitude less accurate than that for the TM polarization. Nonetheless, the TE polarization data also fit linear trend lines with integer slopes when the discretization is fine enough. In order from lowest (2) to highest (12) Nyström orders, the observed slopes are 2, 1, 3, 3, 5, and 5.

Cross section calculations resulting from the second-kind formulation of the TE polarization scattering problem are generally more accurate than those of the first-kind formulation. In fact, as the Nyström order increases, they become nearly as accurate as those for the TM polarization case. Again, the reason is that the singularity of the kernel for the second-kind TE case is no worse than $\log(r)$, which is also the singularity of the kernels in the first and second-kind TM polarization cases.

The process of improving a discretization by reducing the size of the patches is called "h-refinement." This is what has been exhibited in the previous two figures. Keeping the number of patches fixed and increasing the number of parameters used to describe the source distribution on each patch, on the other hand, is known as "p-refinement." With a high-order Nyström code such as FastScat, p-refinement is accomplished by increasing the Nyström order for a given meshing. In general, this is the preferred method for improving a discretization for two reasons: one can avoid the usually tedious process of remeshing the scatterer, and the accuracy of the answer usually improves faster this way. The data in the next plot demonstrate this feature.

Figure 4 presents the TM and TE polarization data given in Figs. 2 and 3 in a different way. The behavior of the calculation for each polarization under p-refinement is illustrated by connecting points corresponding to a fixed number of patches instead of a fixed Nyström order. In some cases, data points corresponding to Nyström orders higher than 12 have been added. The fact that the data points on a semilog plot can be connected by nearly straight lines indicates that p-refinement can achieve *exponential* convergence, as opposed to the *geometric* convergence that was observed for h-refinement. The convergence rate gets higher the larger the patch size.

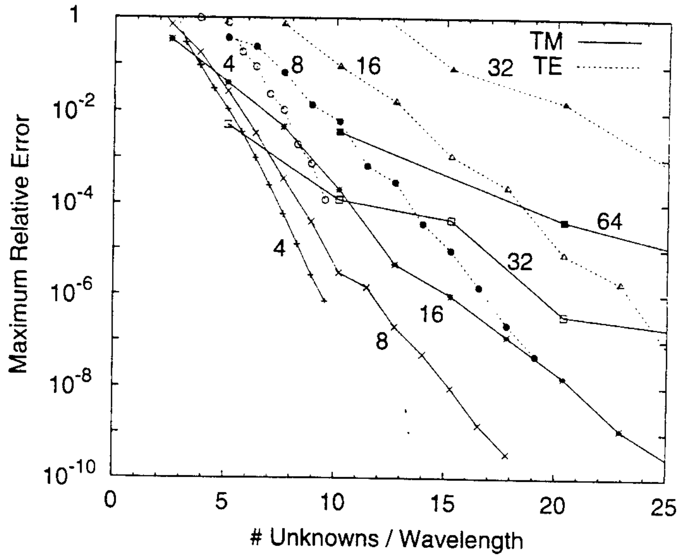


FIG. 4. Semilog plot of maximum relative error vs unknown density for scattering from a 1 λ-radius circle. Points corresponding to different Nyström quadrature orders for a fixed patch size are connected by lines (*solid* for TM polarization and *dashed* for TE polarization) and labeled by the number of patches.

With regard to numbers of unknowns, the most efficient way to achieve high accuracy is to use a high-order method on large patches. For example, with only four patches and a 30th-order quadrature rule, it was possible to achieve an accuracy of 10^{-6} for the TM polarization case and 10^{-4} in the TE case. With this discretization, the unknown density is about 10 unknowns/wavelength and the arc length of each patch is about $1\frac{1}{2}$ wavelengths. For lower accuracies, the advantage of using large patches and high-order methods on the circle is less clear. As a general rule, the optimum discretization is one that uses large patches and high-order methods over smooth regions of the scatterer and smaller patches over more highly curved regions.

A.1.b. 20 λ × 2 λ *ellipse.* A 20 λ × 2 λ ellipse is a 2D scatterer that is less symmetric than a circle, but is still smooth. It is a more challenging scattering problem than a 1 λ-radius circle for several reasons, not least of which is the fact that it extends much more than a wavelength in at least one dimension. In addition, it is a good candidate problem for applying the discretization rule described above.

In our code, the ellipse is described by the pair of parametric equations,

$$\begin{aligned} x &= a\cos u, \\ y &= b\sin u, \end{aligned} \tag{24}$$

where $a = 10\,\lambda$ and $b = 1\,\lambda$. A sensible patching, which puts the highest density of patches in the most highly curved regions and vice versa for the flatter regions, is obtained if the patches cover equal increments in the parameter u. The circumference of a 20 λ × 2 λ ellipse is about 40.64 λ.

We used FastScat to compute the monostatic cross section of a 20 λ × 2 λ ellipse discretized using several different combinations of patch number and Nyström order. The boundary conditions on the surface were either Dirichlet or Neumann, corresponding to TM and TE polarizations, respectively.

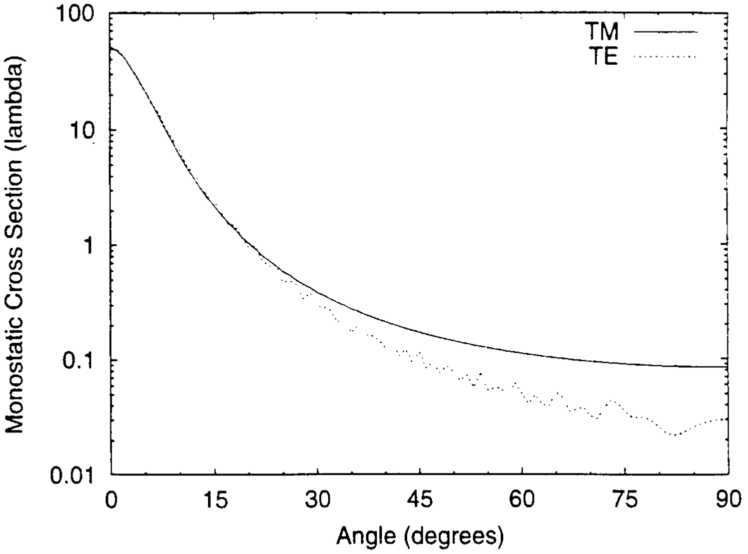


FIG. 5. Monostatic cross section of a 20 λ × 2 λ ellipse for TM and TE polarizations. One quadrant of observation angles is shown; the others may be obtained by considering the fourfold symmetry of the scatterer.

We do not have at our disposal a series solution for the cross section of an ellipse (which we might otherwise use to compute an arbitrarily accurate reference solution). However, we can still estimate the accuracy of the computed solutions by comparing them to the most finely discretized solution, which we designate the "reference solution." We computed reference solutions for the TM and TE polarization cases by meshing the ellipse into 128 patches and putting a 20th-order Gauss-Legendre rule (i.e., 10 sample points) on each patch. We deduce that these reference solutions are accurate to at least six decimal places, given the high-order manner in which all the more coarsely discretized solutions are observed to converge to them. Plots of the monostatic cross section versus incident angle for the reference solutions are given in Fig. 5. As seen in the figure, the monostatic cross section for TM polarization ranges from about 50 λ looking at the broadside to less than 0.1 λ looking at the tip. The TE cross section is similar, although it is not as smooth a function of angle. In both cases, the dynamic range of the cross section is more than 500.

The p-refinement behavior of the calculations on the ellipse using first-kind integral equation formulations for both TM and TE polarization is shown in Fig. 6. Like the circle, exponential convergence is observed and accurate solutions are most efficiently obtained when the mesh consists of patches larger than a wavelength.

A.2. Three-dimensional vector. As in the 2D scalar case, first-kind and second-kind integral formulations were explored. For 3D vector scattering off a PEC scatterer, the first-kind formulation is the electric field integral equation (EFIE) [17]

$$\mathbf{E}^{\mathrm{inc}}_{\mathrm{tan}}(\mathbf{x}) = i\omega \oint_S ds' \left[-G(\mathbf{x},\mathbf{x}')\,\mathbf{J}(\mathbf{x}') + \frac{1}{k^2}\nabla(\nabla' G(\mathbf{x},\mathbf{x}')\cdot\mathbf{J}(\mathbf{x}'))\right]_{\mathrm{tan}}, \qquad (25)$$

and the second-kind formulation is the magnetic field integral equation (MFIE)

$$\mathbf{H}^{\mathrm{inc}}_{\mathrm{tan}}(\mathbf{x}) = -\frac{1}{2}\hat{\mathbf{n}}\times\mathbf{J}(\mathbf{x}) + \oint_S ds'[\nabla' G(\mathbf{x},\mathbf{x}')\times\mathbf{J}(\mathbf{x}')]_{\mathrm{tan}}, \qquad (26)$$

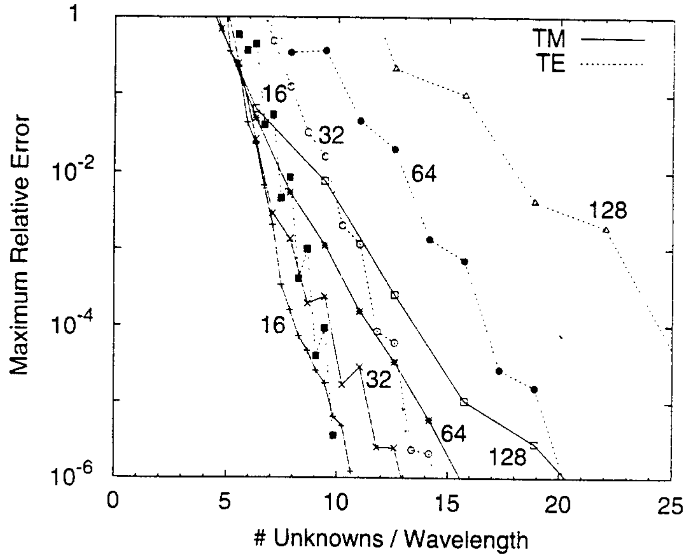


FIG. 6. Semilog plot of maximum relative error vs unknown density for scattering from a 20 λ × 2 λ ellipse. Points corresponding to different Nyström orders for a fixed patch size are connected by lines (*solid* for TM polarization and *dashed* for TE polarization) and labeled by the number of patches.

where $G(\mathbf{x}, \mathbf{x}') \equiv \exp(ik|\mathbf{x}-\mathbf{x}'|)/|\mathbf{x}-\mathbf{x}'|$ is the Helmholtz kernel in 3D, $k = |\mathbf{k}| = \omega/c$ is the radiation wavenumber, $\mathbf{J}$ refers to the electric surface current, $\mathbf{E}^{\text{inc}}$ and $\mathbf{H}^{\text{inc}}$ are the incident electric and magnetic fields, and the subscript *tan* means that only the vector components tangent to surface at the field point are being used.

The EFIE and MFIE can be summed to form a combined field integral equation (CFIE) having some of the same desirable properties as the CFIE in the 2D scalar case. Although no CFIE results are reported in this paper, the same techniques apply.

Note also, that, while the results presented here are restricted to PEC scatterers, it is trivial to generalize the method to the more general scattering problem of homogeneous regions with smooth boundaries.

A.2.a. *One-fourth λ-radius sphere.* Writing a code that correctly calculates 3D vector scattering results is more difficult than writing a correct 2D scalar code. This is doubly true if the code is designed to be high order. Therefore, it is particularly important to verify that the output of a purportedly high-order 3D vector code actually converges to the correct answer under both h- and p-refinement and that it does so in a high-order fashion. In this subsection, we present results demonstrating that our 3D vector Nyström code achieves high-order convergence to the correct answer on a sphere.

A sphere is the ideal surface to use for benchmarking a high-order 3D vector code for the same reasons that a circle is ideal for a high-order 2D scalar code—it is uniformly smooth and the accuracy of computed results can be determined by comparison to the Mie series solution. Since the size of the surface, and therefore the number of unknowns, grows in proportion to r^2 for a sphere, as opposed to just r for a circle, memory limitations prevented us from pushing the unknown density on a 1 λ-radius sphere to the same extremes as were possible on a 1 λ-radius circle. Nonetheless, when we did run FastScat on a 1 λ-radius sphere with a wide selection of discretizations, we found that the results converged to the correct answer just as one would expect for a high-order scattering code. To reach the asymptotic regime, where the convergence behavior is more obvious, however, we chose the radius

TABLE I
3D Quadrature Rule and Testing Function Parameters

Nyström quadrature order	Number sample points	Maximum testing function degree	Number testing functions
2	1	0	1
3	3	1	3
5	6	2	6
7	12	3	10
8	15	4	15

of the sphere to be $\frac{1}{4}\lambda$, which allows us to increase the unknown density fourfold before running out of primary memory (for storing the full impedance matrix). For this reason alone we present the data for the $\frac{1}{4}\lambda$-radius sphere.

The internal surface representation of the sphere corresponds to an ideal sphere and its surface is assumed to be perfectly conducting. The coarsest patching of the sphere consists of 20 identical triangular patches, formed by mapping the triangles of an inscribed icosahedron onto the surface of the sphere. Finer meshes were generated by dividing each of the 20 triangles into n^2 nearly identical subtriangles, where n ranged from 2 up to 10. The distribution of Nyström quadrature points on each patch was determined by a high-order triangle rule [16]. The triangle rule orders that we used and corresponding numbers of sample points are given in Table I. The number of testing functions (products of monomials in the two surface parameters) and the maximum degree of the testing functions used with each triangle rule are also listed in the table.

In all cases except Nyström order 7, the number of sample points equals the number of testing functions, resulting in an exactly-determined local correction linear system. In the seventh-order case, the maximum testing function degree was chosen to make an under-determined linear system.

Solutions for the bistatic cross section of the $\frac{1}{4}\lambda$-radius sphere were computed with the various discretizations and compared against the Mie series solution (shown in Fig. 7). For a sphere this small, the cross sections for the two polarizations are similar (in terms of smoothness and dynamic range), so we present the discretization refinement results only for the $\theta\theta$ case. Cross polarization results are also not presented at all, although it may be noted that such computed cross sections were extremely small (i.e., always less than the co-polarized results by at least eight orders of magnitude).

The convergence behavior of the scattering results under h-refinement is shown in Fig. 8. Refining the mesh for a given Nyström order always improves the accuracy of the solution. It is apparent for the lower Nyström orders that the data approach linear trend lines with integer slopes as the patches get smaller, just as they did in 2D. In the case of the EFIE, the slopes of the trend lines for Nyström orders 2 and 3 are both unity and in the case of the MFIE, they are 2 and 3, respectively. For the higher orders, the slopes appear to be increasing, but it is not as clear what their asymptotic values will be. For Nyström order 5, the last pair of points produce slopes close to 3 and 5 for the EFIE and MFIE solutions, respectively. In all cases, the solution at a particular discretization obtained by using the less singular kernel (i.e., the MFIE) is more accurate.

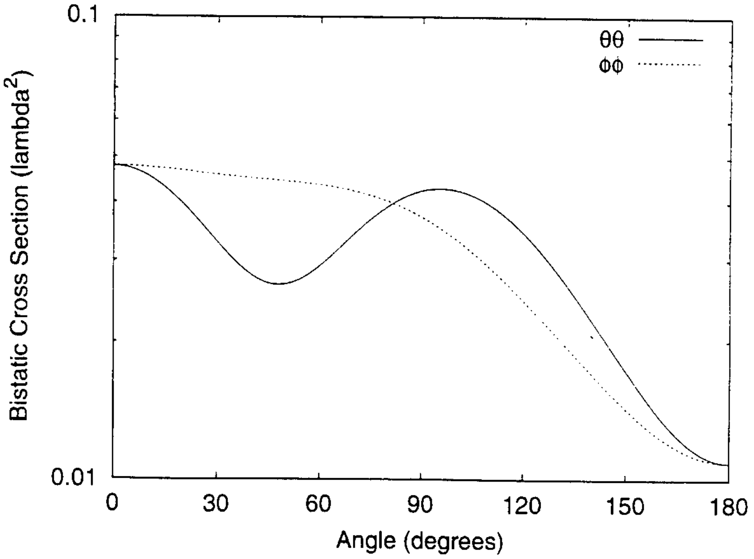


FIG. 7. Bistatic cross section of a $\frac{1}{4}$ λ-radius PEC sphere for $\theta\theta$ and $\phi\phi$ polarizations computed by the Mie series.

The behavior of the sphere results under p-refinement are shown in Fig. 9. The observed p-refinement behavior is similar to that in the 2D scalar case. The fastest convergence is usually achieved by applying a high-order quadrature to a coarse meshing. One notable difference from the 2D scalar case is that the 3D vector calculation requires a higher density of unknowns to achieve a comparable maximum relative error in the bistatic cross section. The jaggedness of the p-refinement curves for the EFIE data may be explained by reference to the h-refinement plot, which shows that the 2nd- and 3rd-order results have nearly the same accuracy, and that the 7th-order results are actually less accurate than those for 5th-order.

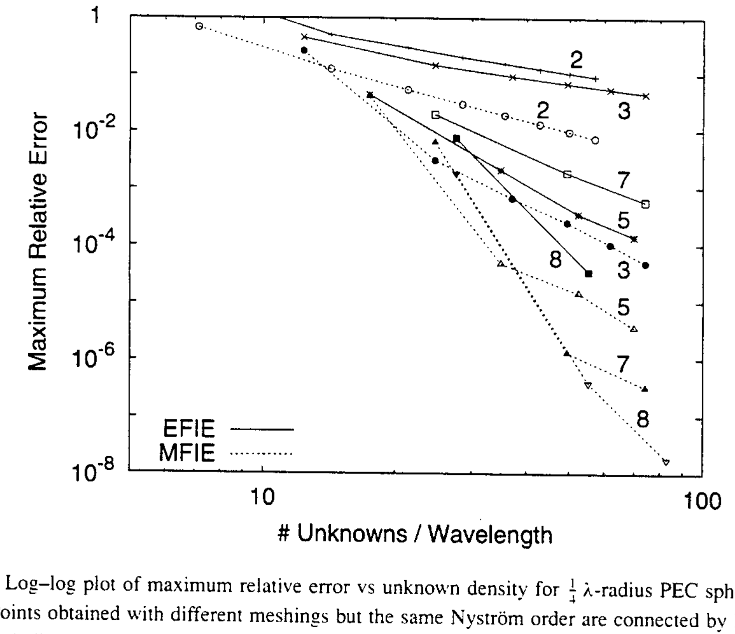


FIG. 8. Log–log plot of maximum relative error vs unknown density for $\frac{1}{4}$ λ-radius PEC sphere in $\theta\theta$ polarization. Points obtained with different meshings but the same Nyström order are connected by lines. A solid (dashed) line indicates use of the EFIE (MFIE) integral formulation.

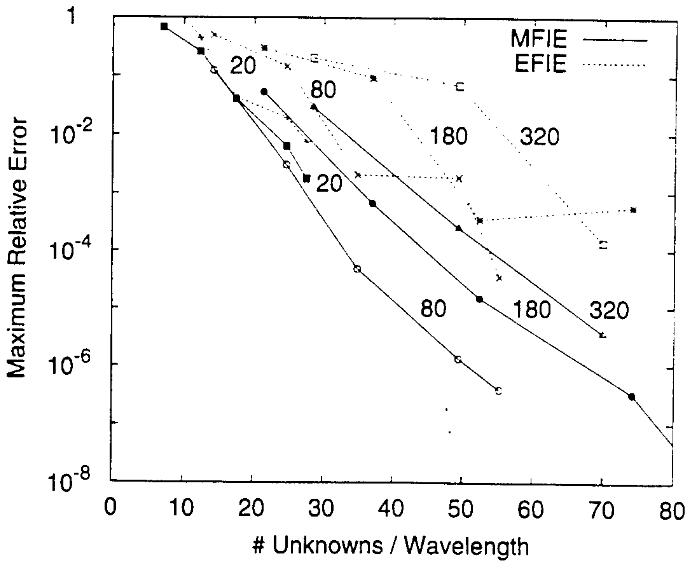


FIG. 9. Semilog plot of maximum relative error vs unknown density for scattering from a $\frac{1}{4}$ λ-radius PEC sphere. Points corresponding to different Nyström quadrature orders for a fixed patch size are connected by lines (*solid* for MFIE and *dashed* for EFIE) and labeled by the number of patches.

For Nyström orders higher than about 8, problems related to ill-conditioning arise in the EFIE formulation. Although the increasingly ill-conditioned nature of the local correction linear system is a contributing factor, the more important contribution probably comes from the fact that the EFIE is especially susceptible to conditioning problems when the Nyström sample points get too close together. Unfortunately, this is exactly what happens for the higher-order triangle rules. As the order increases, the quadrature points tend to bunch up near the edges and corners of the triangle. It may be possible to overcome this problem by inventing different high-order triangle rules with better sample point spacing and by using a better conditioned integral equation formulation such as the MFIE or CFIE (combined field integral equation).

A.2.b. $2\ \lambda \times 2\ \lambda \times 0.2\ \lambda$ *ellipsoid.* As an example of a smooth, but less symmetric 3D scatterer, we next consider a PEC ellipsoid with principal axis diameters 2 λ, 2 λ, and 0.2 λ. We computed the monostatic cross sections of this discus-shaped scatterer in $\theta\theta$ and $\phi\phi$ polarizations using a MFIE formulation and an eighth-order quadrature rule, which put 15 points on each patch. Four different meshings, comprising 20, 80, 180, and 320 patches, were tried. Each meshing was tailored to put smaller patches in the vicinity of the $r = 1\ \lambda$ equator, where the one of the radii of curvature is small, and larger patches everywhere else, where the surface is relatively flat. The number of unknowns distributed over the 6.47 λ^2 surface of the ellipsoid in the four cases ranged from 600 with the coarsest meshing to 9600 with the finest.

As we did with the ellipse in 2D, we can designate the solution computed with the finest discretization to be the reference solution and obtain accuracy estimates of the other solutions by comparing them to this reference solution. Figure 10 shows the reference solutions for the $\theta\theta$ and $\phi\phi$ polarization cases.

Differences between the reference solution and the other, less finely discretized solutions are shown in Fig. 11. As expected, the accuracy of the solution improves as one refines the

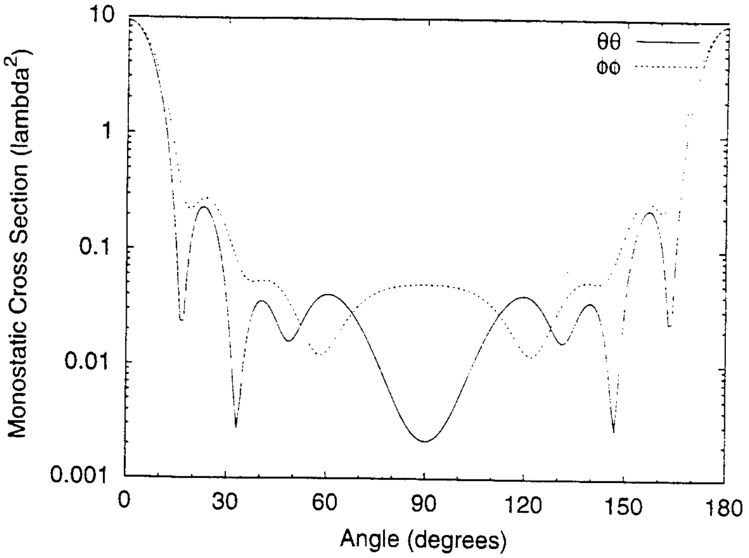


FIG. 10. Reference solutions for the monostatic cross section of a 2 λ × 2 λ × 0.2 λ PEC ellipsoid in $\theta\theta$ and $\phi\phi$ polarizations. At 0° the observer is looking at the flattest part of the ellipsoid; at 90° he is looking edge on.

discretization. It should also come as no surprise that the solutions are also most accurate near 0° and 180°, where the cross section is highest. What is particularly notable about this plot, however, is the fact that the error in the cross section decreases by orders of magnitude when one reduces the (linear) size of each patch by factors of 2 or 3. Such large reductions in the error are a direct consequence of our using an exact surface description and a high-order rule (8th-order, in this case) on each patch.

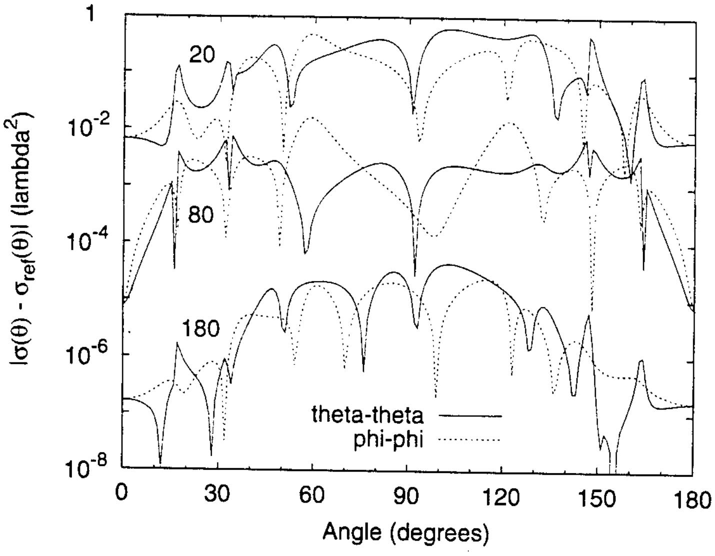


FIG. 11. Semilog plot of the differences between cross sections computed using meshings consisting of 20, 80, and 180 patches, and a reference cross section computed using a meshing consisting of 320 patches. The asymmetry of each curve reflects the fact that the meshings did not possess reflection symmetry.

B. *Run-Time Performance Comparisons*

In this section we compare the run-time performance of our high-order Nyström implementation of FastScat to that of two method of moments scattering codes. The first comparison code is an earlier, high-order Galerkin implementation of FastScat [20]. The second is a low-order code (RWG basis and testing functions on flat facets) called FISC [21]. We ran each code under comparable conditions to obtain solutions for the bistatic cross section in the $\theta\theta$ polarization of three different size PEC spheres. The high-order Nyström discretizations were constructed using an eighth-order quadrature rule (15 sample points per patch) and fourth-degree testing functions for computing local corrections. The high-order Galerkin discretizations were constructed from the same surface mesh using patch-based, polynomial (in the parameterization) basis functions up to degree 4 to give the same number of unknowns per patch, namely 30. The surface mesh used by FISC was necessarily different from that used by both versions of FastScat because, with an RWG discretization, one unknown is associated with each edge rather than multiple unknowns being associated with each patch. Nonetheless, its surface meshes were constructed to maintain the density of unknowns at about 7.7 unknowns/wavelength, the same as for the both FastScat discretizations. All computations were performed using a dense matrix fill, an LUD solver, and a MFIE formulation.

Table II gives a summary of the results. The reported times are run times on a SPARC-10 workstation with 512 MB primary memory. The total run time is broken into setup time (which includes the time spent setting up the problem and filling the impedance matrix) and solve time (which includes the time spent performing the LUD and solving for the bistatic cross section at 181 angles).

In comparing the results from the two high-order implementations of FastScat, two features are especially noteworthy. The first is that the high-order Galerkin result is more accurate by about a factor of 5 than the high-order Nyström result. The second is that use of the Nyström discretization can speed up the setup phase of the computation enormously, with the speedup factor increasing as the number of unknowns increases. The observation that the high-order Galerkin code computes results somewhat more accurately than the Nyström code is consistent with our experience computing cross sections for other scatterers, both in 2D and 3D. It is compensated, however, by the fact that the setup phase (and to a lesser extent the solve phase) runs much faster using the Nyström code. Furthermore, the factor of 5 difference in accuracy is actually less significant in this case than it would

TABLE II
Nyström vs Galerkin Performance on PEC Spheres

Scattering code	Radius (λ)	No. of unknowns	Setup time (s)	Solve time (s)	RMS error (dB)
FastScat (Nyström)	0.9	600	74	36	0.35
FastScat (Galerkin)	0.9	600	972	88	0.07
FISC (Galerkin)	0.9	600	83	42	1.28
FastScat (Nyström)	1.8	2400	539	2742	0.26
FastScat (Galerkin)	1.8	2400	8177	3395	0.05
FISC (Galerkin)	1.8	2430	873	2255	0.61
FastScat (Nyström)	2.7	5400	1953	31735	0.097
FastScat (Galerkin)	2.7	5400	38803	36152	0.021
FISC (Galerkin)	2.7	5880	8230	28795	0.723

be if we were comparing low-order codes. Given the $\mathcal{O}(h^9)$ convergence rate expected of an eighth-order quadrature rule, it should be possible to recover the factor of 5 in accuracy with further h-refinement by a modest 20%.

The high-order Nyström code computes more accurate answers than the low-order Galerkin code (FISC) in all cases. For the spheres considered here, this is largely due to the fact that FISC uses a low-order surface representation. The high-order Nyström code also requires less setup time, an advantage that grows as the problems get bigger. Even a comparison based on total solution time shows the high-order Nyström implementation of FastScat to be more efficient for computing accurate answers.

Finally, it is useful to note that an equivalent Nyström discretization exists for every method of moments discretization and vice versa [22], so it is possible, at least in principle, to eliminate the observed accuracy discrepancy between the two versions of FastScat by implementing a Nyström code whose discretization error precisely matches that obtained by the Galerkin code. We have not attempted to do this, but suspect that to do so would entail additional complications and computations that would negate the substantial simplicity and efficiency of the present implementation. On balance, we find the high-order Nyström method in its present form preferable to the high-order Galerkin method for solving integral equations, especially when one adds in its other benefits such as reduced implementation complexity and potential for significantly improved FMM performance.

V. SUMMARY

The standard Nyström method is a simple and efficient mechanism for discretizing integral equations. We have shown how it can be adapted to provide a high-order discretization of the boundary integral equations of wave scattering in 2D and 3D, which have singular kernels. Numerical results obtained with a software implementation of this method show that the algorithm can achieve high-order convergence to the correct answer for scattering cross sections in 2D and 3D. We also demonstrated that a high-order Nyström code considerably reduces the CPU time cost of a scattering calculation by comparison to a high-order Galerkin code, especially the precomputation time cost. The high-order Nyström code also outperformed a well-tuned, low-order Galerkin code (FISC) in terms of solution accuracy and total run time. Demonstrations of how a high-order Nyström code can be used in conjunction with the FMM to reduce the memory and CPU time requirements of solving large scattering problems will be the subject of a future publication.

APPENDIX

A. *Local Corrections*

Eleven different kernels arise in boundary integral equation formulations of 2D scalar, 3D scalar, and 3D electromagnetic scattering:

2D & 3D Scalar	3D Electromagnetic
$G(r)$ $\hat{\mathbf{n}}' \cdot \nabla' G(r)$ $\hat{\mathbf{n}} \cdot \nabla G(r)$ $(\hat{\mathbf{n}} \cdot \nabla)(\hat{\mathbf{n}}' \cdot \nabla' G(r))$	$G(r)(\mathbf{t}(\mathbf{x}) \cdot \mathbf{t}'(\mathbf{x}'))$ $\mathbf{t}(\mathbf{x}) \cdot (\nabla' G(r) \times \mathbf{t}'(\mathbf{x}'))$ $(\mathbf{t}(\mathbf{x}) \cdot \nabla)(\nabla' G(r) \cdot \mathbf{t}'(\mathbf{x}'))$

where

$$G(r) \equiv \begin{cases} \frac{i}{4} H_0^{(1)}(kr) & \text{in 2D,} \\ \frac{e^{ikr}}{r} & \text{in 3D,} \end{cases} \tag{27}$$

r is the magnitude of the vector $\mathbf{r} \equiv \mathbf{x}' - \mathbf{x}$ from the field point at $\mathbf{x}$ to the source point at $\mathbf{x}'$; k is the wavenumber of the waves; $\hat{\mathbf{n}}$ and $\hat{\mathbf{n}}'$ are the unit normals to the surface at the field and source points, respectively; ∇ and ∇' are gradient operators for the field and source coordinates, respectively; and $H_0^{(1)}$ refers to the zeroth order Hankel function of the first kind, defined by $H_0^{(1)}(x) \equiv J_0(x) + iY_0(x)$, where $J_n(x)$ and $Y_n(x)$ represent nth-order Bessel functions of the first and second kinds, respectively.

For the 3D electromagnetic case, the source and excitation are surface tangent vectors so it becomes necessary to compute local corrections for four scalar kernels, one for each of the four combinations of (two) independent surface tangent vectors at the field point and (two) independent surface tangent vectors at the source point. These surface tangent vectors at the field and source points, represented by $\mathbf{t}(\mathbf{x})$ and $\mathbf{t}'(\mathbf{x}')$, respectively, are included as part of the 3D electromagnetic kernel in recognition of this fact and for clarity of presentation.

In this section, we show how to compute local corrections for each of these kernels. We will make use of the vector calculus identity [23]

$$(\hat{\mathbf{n}} \cdot \nabla)(\hat{\mathbf{n}}' \cdot \nabla' g(r)) = (\hat{\mathbf{n}} \cdot \hat{\mathbf{n}}')(\nabla \cdot \nabla' g(r)) - (\hat{\mathbf{n}} \times \nabla) \cdot (\hat{\mathbf{n}}' \times \nabla' g(r)) \tag{28}$$

$$= (\hat{\mathbf{n}} \cdot \hat{\mathbf{n}}') k^2 g(r) - (\hat{\mathbf{n}} \times \nabla) \cdot (\hat{\mathbf{n}}' \times \nabla' g(r)), \tag{29}$$

where the second line follows if $g(r)$ obeys the homogeneous Helmholtz equation

$$(\nabla^2 + k^2) g(r) = 0. \tag{30}$$

This identity allows one to convert between double normal derivative and double tangential derivative operators on the Green function.

A.1. Two-dimensional scalar.

A.1.a. $G(r)$,

$$G(r) = \frac{i}{4} H_0^{(1)}(kr) = \underbrace{\frac{i}{4} J_0(kr)}_{\text{regular}} - \underbrace{\frac{1}{4} Y_0(kr)}_{\text{singular}}. \tag{31}$$

This kernel may be written as the sum of a regular part and a singular part. It is necessary to compute local corrections only for the singular part because the regular part will be efficiently integrated by the underlying high-order quadrature rule. The function $Y_0(kr)$ contains a $\log(r)$ singularity. Therefore, one can use "lin–log" quadrature rules [24] to efficiently compute local correction integrals when the region of integration contains the field point, and Gauss–Legendre rules otherwise.

A.1.b. $\hat{\mathbf{n}}' \cdot \nabla' G(r)$,

$$\hat{\mathbf{n}}' \cdot \nabla' G(r) = \frac{\hat{\mathbf{n}}' \cdot \mathbf{r}}{r} \frac{d}{dr} G(r) = \underbrace{-\frac{i}{4} k^2 \overbrace{(\hat{\mathbf{n}}' \cdot \mathbf{r})}^{\text{regular}} \overbrace{\frac{J_1(kr)}{kr}}^{\text{regular}}}_{\text{regular}} + \underbrace{\frac{1}{4} \overbrace{\frac{\hat{\mathbf{n}}' \cdot \mathbf{r}}{r^2}}^{\text{regular}} \overbrace{kr Y_1(kr)}^{\text{singular}}}_{\text{singular}}. \tag{32}$$

The first term is regular; the second is singular. The second term is singular not because its value diverges at the origin (in fact, $\lim_{r\to 0}(\hat{\mathbf{n}}' \cdot \mathbf{r}/r^2)krY_1(kr) = 1/\pi R$, where R is the radius of curvature of the surface at the field point), but because its higher derivatives do. The singularity is still a $\log(r)$ singularity, so local correction integrals can be computed in the same manner as for the previous kernel.

A.1.c. $\hat{\mathbf{n}} \cdot \nabla G(r)$.

$$\hat{\mathbf{n}} \cdot \nabla G(r) = -\frac{\hat{\mathbf{n}} \cdot \mathbf{r}}{r}\frac{d}{dr}G(r) = \underbrace{\frac{i}{4}k^2\overbrace{(\hat{\mathbf{n}} \cdot \mathbf{r})}^{\text{regular}}\overbrace{\frac{J_1(kr)}{kr}}^{\text{regular}}}_{\text{regular}} - \underbrace{\frac{1}{4}\overbrace{\frac{\hat{\mathbf{n}} \cdot \mathbf{r}}{r^2}}^{\text{regular}}\overbrace{krY_1(kr)}^{\text{singular}}}_{\text{singular}}. \tag{33}$$

This kernel is identical to that for $\hat{\mathbf{n}}' \cdot \nabla' G(r)$ with $\hat{\mathbf{n}}'$ replaced by $-\hat{\mathbf{n}}$ and it has similar properties.

A.1.d. $(\hat{\mathbf{n}} \cdot \nabla)(\hat{\mathbf{n}}' \cdot \nabla' G(r))$.

$$\begin{aligned}(\hat{\mathbf{n}} \cdot \nabla)&(\hat{\mathbf{n}}' \cdot \nabla' G(r)) \\ &= \frac{(\hat{\mathbf{n}} \cdot \mathbf{r})(\hat{\mathbf{n}}' \cdot \mathbf{r})}{r^2}\left(\frac{1}{r}\frac{dG(r)}{dr} - \frac{d^2G(r)}{dr^2}\right) - \frac{(\hat{\mathbf{n}} \cdot \hat{\mathbf{n}}')}{r}\frac{dG(r)}{dr} \end{aligned}\tag{34}$$

$$\begin{aligned}&= \underbrace{\frac{ik^2}{4}\left(\overbrace{(\hat{\mathbf{n}} \cdot \hat{\mathbf{n}}')}^{\text{regular}}\overbrace{\frac{J_1(kr)}{kr}}^{\text{regular}} - \overbrace{\frac{(\hat{\mathbf{n}} \cdot \mathbf{r})(\hat{\mathbf{n}}' \cdot \mathbf{r})}{r^2}}^{\text{regular}}\overbrace{J_2(kr)}^{\text{regular}}\right)}_{\text{regular}} \\ &\quad + \underbrace{(\hat{\mathbf{n}} \cdot \nabla)(\hat{\mathbf{n}}' \cdot \nabla' G^R(r))}_{\text{hypersingular}}.\end{aligned}\tag{35}$$

Applying the derivatives to the real part of $G(r)$, namely $G^R(r) \equiv -\frac{1}{4}Y_0(kr)$, produces a term that is not merely singular but hypersingular. When convolved with a regular function, this term is not (in general) integrable because it diverges like $1/r^2$, relative to the field point. The following discussion shows how to manipulate it into a form that allows numerical evaluation when the region of integration contains the field point. When the region of integration does not include the field point, Gauss–Legendre rules may be used.

The convolution of $(\hat{\mathbf{n}} \cdot \nabla)(\hat{\mathbf{n}}' \cdot \nabla' G^R(r))$ with testing function $f(\mathbf{x}')$ is

$$\int_C dl'(\hat{\mathbf{n}} \cdot \nabla)(\hat{\mathbf{n}}' \cdot \nabla' G^R(r))f(\mathbf{x}'). \tag{36}$$

Strictly speaking this is not a proper integral unless it is assumed to represent the limiting value as the field point approaches the surface from off the surface. We implicitly make this assumption throughout. Using the vector identity (29) and the fact that $G^R(r)$ obeys the homogenous Helmholtz equation when $\mathbf{x}$ is not on S, we can convert the double normal derivative operator to a double tangential derivative operator:

$$\int_C dl'[k^2(\hat{\mathbf{n}} \cdot \hat{\mathbf{n}}')G^R(r) - (\hat{\mathbf{n}} \times \nabla) \cdot (\hat{\mathbf{n}}' \times \nabla' G^R(r))]f(\mathbf{x}'). \tag{37}$$

In 2D, we can rewrite the second term even more explicitly in terms of tangential derivatives, obtaining

$$\int_C dl'[k^2(\hat{\mathbf{n}}\cdot\hat{\mathbf{n}}')G^R(r) - (\hat{\mathbf{t}}\cdot\nabla)(\hat{\mathbf{t}}'\cdot\nabla' G^R(r))]f(\mathbf{x}'), \tag{38}$$

where $\hat{\mathbf{t}}$ and $\hat{\mathbf{t}}'$ are unit tangent vectors at the field and source points, respectively. The first term has a $\log(r)$ singularity, which we already know how to integrate numerically; the second term is hypersingular and requires further manipulation.

The gradient operators ∇ and ∇' commute with the unit tangent vectors $\hat{\mathbf{t}}'$ and $\hat{\mathbf{t}}$, respectively, so we can rearrange the factors of the second term and integrate it by parts as

$$\begin{aligned} -\int_C dl'(\hat{\mathbf{t}}\cdot\nabla)(\hat{\mathbf{t}}'\cdot\nabla' G^R(r))f(\mathbf{x}') \\ = -\int_C dl' f(\mathbf{x}')\hat{\mathbf{t}}'\cdot\nabla'(\hat{\mathbf{t}}\cdot\nabla G^R(r)) \qquad (39) \\ = -\int_C dl'\hat{\mathbf{t}}'\cdot\nabla'(f(\mathbf{x}')(\hat{\mathbf{t}}\cdot\nabla G^R(r))) \\ + \int_C dl'(\hat{\mathbf{t}}'\cdot\nabla' f(\mathbf{x}'))(\hat{\mathbf{t}}\cdot\nabla G^R(r)). \qquad (40) \end{aligned}$$

The first integral on the right-hand side of (40) is

$$\begin{aligned} -\int_C dl'\hat{\mathbf{t}}'\cdot\nabla'(f(\mathbf{x}')(\hat{\mathbf{t}}\cdot\nabla G^R(r))) \\ = -\int_C d\mathbf{l}'\cdot\nabla'(f(\mathbf{x}')(\hat{\mathbf{t}}\cdot\nabla G^R(r))) \qquad (41) \\ = -[f(\mathbf{x}')(\hat{\mathbf{t}}\cdot\nabla G^R(r))]_{C_1}^{C_2}; \qquad (42) \end{aligned}$$

i.e., since the integrand is a total derivative, the value of the integral is a difference of values at the endpoints. Rearranging factors and using

$$\nabla G^R(r) = -\nabla' G^R(r), \tag{43}$$

we can rewrite the second integral as

$$-\int_C dl'\,\nabla' G^R(r)\cdot[\hat{\mathbf{t}}(\hat{\mathbf{t}}'\cdot\nabla' f(\mathbf{x}'))]. \tag{44}$$

In this form, the integral is not yet evaluable because $\nabla' G^R(r)$ diverges like $1/r$ relative to the field point. We can make it integrable by adding and subtracting a smooth function that matches the integrand at the field point. Specifically, let us write (44) as

$$-\int_C dl'\,\nabla' G^R(r)\cdot[\hat{\mathbf{t}}(\hat{\mathbf{t}}'\cdot\nabla' f(\mathbf{x}')) - \hat{\mathbf{t}}'(\hat{\mathbf{t}}\cdot\nabla' f(\mathbf{x}))] - \int_C dl'\,\nabla' G^R(r)\cdot[\hat{\mathbf{t}}'(\hat{\mathbf{t}}\cdot\nabla' f(\mathbf{x}))]. \tag{45}$$

where $\hat{\mathbf{t}}' \cdot \nabla' f(\mathbf{x}')$ and $\hat{\mathbf{t}} \cdot \nabla' f(\mathbf{x})$ represent tangential derivatives of the testing function $f(\mathbf{x}')$ evaluated at the field and source points, respectively. The first integral in this expression is integrable because the zero of

$$[\hat{\mathbf{t}}(\hat{\mathbf{t}}' \cdot \nabla' f(\mathbf{x}')) - \hat{\mathbf{t}}'(\hat{\mathbf{t}} \cdot \nabla' f(\mathbf{x}))] \tag{46}$$

at the field point cancels the pole from $\nabla' G^R(r)$ at the field point, leaving a singularity no worse than $\log(r)$ relative to the field point. By rearranging factors, the integrand of the second integral can be shown to be a total derivative, so that

$$\begin{aligned} -\int_C dl'\, \nabla' G^R(r) \cdot [\hat{\mathbf{t}}'(\hat{\mathbf{t}} \cdot \nabla' f(\mathbf{x}))] \\ = -\int_C dl'\, \hat{\mathbf{t}}' \cdot (\nabla' G^R(r)(\hat{\mathbf{t}} \cdot \nabla' f(\mathbf{x}))) & \quad (47) \\ = -[G^R(r)(\hat{\mathbf{t}} \cdot \nabla' f(\mathbf{x}))]_{C_1}^{C_2}. & \quad (48) \end{aligned}$$

Putting the various terms together, we arrive at the following numerically tractable expression for the integral needed to compute local corrections for the hypersingular component of the kernel

$$\begin{aligned} \int_C dl' \{k^2(\hat{\mathbf{n}} \cdot \hat{\mathbf{n}}') G^R(r) f(\mathbf{x}') - \nabla' G^R(r) \cdot [\hat{\mathbf{t}}(\hat{\mathbf{t}}' \cdot \nabla' f(\mathbf{x}')) - \hat{\mathbf{t}}'(\hat{\mathbf{t}} \cdot \nabla' f(\mathbf{x}))]\} \\ - [f(\mathbf{x}')(\hat{\mathbf{t}} \cdot \nabla G^R(r)) + G^R(r)(\hat{\mathbf{t}} \cdot \nabla' f(\mathbf{x}))]_{C_1}^{C_2}. \end{aligned} \tag{49}$$

or, substituting for $G^R(r)$,

$$\begin{aligned} -\frac{k^2}{4} \int_C dl' \left\{ (\hat{\mathbf{n}} \cdot \hat{\mathbf{n}}') Y_0(kr) f(\mathbf{x}') + \frac{Y_1(kr)}{kr} \hat{\mathbf{r}} \cdot \left[\hat{\mathbf{t}} \frac{df}{dl'}(\mathbf{x}') - \hat{\mathbf{t}}' \frac{df}{dl'}(\mathbf{x}) \right] \right\} \\ - \frac{1}{4} \left[k^2 \frac{Y_1(kr)}{kr} (\hat{\mathbf{t}} \cdot \hat{\mathbf{r}}) f(\mathbf{x}') - Y_0(kr) \frac{df}{dl'}(\mathbf{x}) \right]_{C_1}^{C_2}. \end{aligned} \tag{50}$$

A.2. Three-dimensional scalar.

A.2.a. $G(r)$.

$$G(r) = \frac{e^{ikr}}{r} = i \underbrace{\frac{\sin(kr)}{r}}_{\text{regular}} + \underbrace{\frac{\cos(kr)}{r}}_{\text{singular}}. \tag{51}$$

As in the 2D scalar case, this kernel may be written as the sum of a regular part and a singular part. It is necessary to compute local corrections only for the singular part because the regular part will be efficiently integrated by the underlying high-order quadrature rule. The singular term contains a $1/r$ singularity. Computing local corrections for the singular part requires evaluation of integrals of $\cos(kr)/r$ times polynomials in the parameters $\mathbf{u} = (u^1, u^2)$ used to describe the surface. When the region of integration contains the field point, it may be subdivided into triangles with the field point at one vertex, and the integration may be performed by using the Duffy transformation [25] and Gauss–Legendre product rules on the subtriangles. Otherwise, one can apply efficient quadrature rules for smooth functions such as high-order triangle rules [16].

A.2.b. $\hat{\mathbf{n}}' \cdot \nabla' G(r)$.

$$\hat{\mathbf{n}}' \cdot \nabla' G(r) = \frac{\hat{\mathbf{n}}' \cdot \mathbf{r}}{r} \frac{d}{dr} G(r) = \frac{(ikr - 1)\, e^{ikr}}{r^2} \frac{\hat{\mathbf{n}}' \cdot \mathbf{r}}{r} \tag{52}$$

$$= \underbrace{ik^3 \overbrace{\frac{\left(\cos(kr) - \frac{\sin(kr)}{kr}\right)}{(kr)^2}}^{\text{regular}} \overbrace{(\hat{\mathbf{n}}' \cdot \mathbf{r})}^{\text{regular}}}_{\text{regular}} - \underbrace{\overbrace{(\cos(kr) + (kr)\sin(kr))}^{\text{regular}} \overbrace{\frac{(\hat{\mathbf{n}}' \cdot \mathbf{r})}{r^2} \frac{1}{r}}^{\text{singular}}}_{\text{singular}}. \tag{53}$$

In 2D, $(\hat{\mathbf{n}}' \cdot \mathbf{r})/r^2$ is a regular function with a removable singularity at the origin. In 3D, the singularity is removable only if the principal radii of curvature of the surface at the field point are the same. Otherwise its limiting value depends on the direction from which the origin is approached. Nonetheless, local correction integrals can be computed efficiently by means of triangle subdivision and the Duffy transformation.

A.2.c. $\hat{\mathbf{n}} \cdot \nabla G(r)$.

$$\hat{\mathbf{n}} \cdot \nabla G(r) = \underbrace{-ik^3 \overbrace{\frac{\left(\cos(kr) - \frac{\sin(kr)}{kr}\right)}{(kr)^2}}^{\text{regular}} \overbrace{(\hat{\mathbf{n}} \cdot \mathbf{r})}^{\text{regular}}}_{\text{regular}} + \underbrace{\overbrace{(\cos(kr) + (kr)\sin(kr))}^{\text{regular}} \overbrace{\frac{(\hat{\mathbf{n}} \cdot \mathbf{r})}{r^2} \frac{1}{r}}^{\text{singular}}}_{\text{singular}}. \tag{54}$$

This kernel is identical to that for $\hat{\mathbf{n}}' \cdot \nabla' G(r)$ with $\hat{\mathbf{n}}'$ replaced by $-\hat{\mathbf{n}}$ and has similar properties.

A.2.d. $(\hat{\mathbf{n}} \cdot \nabla)(\hat{\mathbf{n}}' \cdot \nabla' G(r))$.

$$(\hat{\mathbf{n}} \cdot \nabla)(\hat{\mathbf{n}}' \cdot \nabla' G(r))$$

$$= (\hat{\mathbf{n}} \cdot \hat{\mathbf{n}}') \left(\frac{1 - ikr}{r^3} \right) e^{ikr} + (\hat{\mathbf{n}} \cdot \mathbf{r})(\hat{\mathbf{n}}' \cdot \mathbf{r}) \left(\frac{k^2 r^2 + 3ikr - 3}{r^5} \right) e^{ikr} \tag{55}$$

$$= \underbrace{ik^3 \left(\overbrace{\frac{\left(\frac{\sin(kr)}{kr} - \cos(kr)\right)}{(kr)^2}}^{\text{regular}} \overbrace{(\hat{\mathbf{n}} \cdot \hat{\mathbf{n}}')}^{\text{regular}} + k^2 \overbrace{\frac{\frac{\sin(kr)}{kr} - 3\left(\frac{\frac{\sin(kr)}{kr} - \cos(kr)}{(kr)^2}\right)}{(kr)^2}}^{\text{regular}} \overbrace{(\hat{\mathbf{n}} \cdot \mathbf{r})(\hat{\mathbf{n}}' \cdot \mathbf{r})}^{\text{regular}} \right)}_{\text{regular}}$$

$$+ \underbrace{(\hat{\mathbf{n}} \cdot \nabla)(\hat{\mathbf{n}}' \cdot \nabla' G^R(r))}_{\text{hypersingular}}. \tag{56}$$

Applying the derivatives to the real part of $G(r)$, namely $G^R(r) \equiv \cos(kr)/r$, produces a term that is not merely singular but hypersingular. When convolved with a regular function, this term is not (in general) integrable because it diverges like $1/r^3$ relative to the field point. The following discussion shows how to manipulate it into a form that allows numerical evaluation when the region of integration contains the field point. When the region of integration does not include the field point, standard, high-order rules for integrating regular, two-parameter functions may be used.

The convolution of $(\hat{\mathbf{n}} \cdot \nabla)(\hat{\mathbf{n}}' \cdot \nabla' G^R(r))$ with testing function $f(\mathbf{x}')$ is

$$\int_S ds' (\hat{\mathbf{n}} \cdot \nabla)(\hat{\mathbf{n}}' \cdot \nabla' G^R(\mathbf{x}, \mathbf{x}'))f(\mathbf{x}') \tag{57}$$

or

$$\int_S ds' [k^2 (\hat{\mathbf{n}} \cdot \hat{\mathbf{n}}') G^R(\mathbf{x}, \mathbf{x}') - (\hat{\mathbf{n}} \times \nabla) \cdot (\hat{\mathbf{n}}' \times \nabla' G^R(\mathbf{x}, \mathbf{x}'))] f(\mathbf{x}'), \tag{58}$$

where the second form follows from Eq. (29). As in the 2D case, we implicitly assume a limiting procedure whereby the field point approaches its final destination on the surface from off the surface. The first term in brackets is only singular like $1/r$; we already know how to deal with such expressions. It is the second term that requires further attention. Write this term in component form using the Levi–Civita tensor ϵ_{ijk} and manipulate the expression as shown using the fact that $\mathbf{x}$ and $\mathbf{x}'$ are independent. Summation over repeated indices is implied.

$$\begin{aligned}
-\int_S ds' ((\hat{\mathbf{n}} \times \nabla) \cdot (\hat{\mathbf{n}}' \times \nabla' G^R(\mathbf{x}, \mathbf{x}')))f(\mathbf{x}') \\
&= -(\hat{\mathbf{n}} \times \nabla) \cdot \int_S ds' (\hat{\mathbf{n}}' \times \nabla' G^R(\mathbf{x}, \mathbf{x}'))f(\mathbf{x}') && (59) \\
&= -\epsilon_{ijk} n_j \partial_k \left[\int_S ds' (\hat{\mathbf{n}}' \times \nabla' G^R(\mathbf{x}, \mathbf{x}'))f(\mathbf{x}') \right]_i && (60) \\
&= -\epsilon_{ijk} n_j \left[\int_S ds' \big(\hat{\mathbf{n}}' \times \nabla' (\partial_k G^R(\mathbf{x}, \mathbf{x}'))\big) f(\mathbf{x}') \right]_i && (61) \\
&= -\epsilon_{ijk} n_j \left[\int_S ds'\, \hat{\mathbf{n}}' \times \nabla' \big(f(\mathbf{x}') \partial_k G^R(\mathbf{x}, \mathbf{x}')\big) \right]_i \\
&\quad + \epsilon_{ijk} n_j \left[\int_S ds'\, \partial_k G^R(\mathbf{x}, \mathbf{x}') (\hat{\mathbf{n}}' \times \nabla' f(\mathbf{x}')) \right]_i && (62)
\end{aligned}$$

The last step shows the result of integrating by parts. Letting

$$\psi = f(\mathbf{x}') \partial_k G^R(\mathbf{x}, \mathbf{x}'), \tag{63}$$

we apply an adjunct to Stokes's theorem,

$$\int_S ds (\hat{\mathbf{n}} \times \nabla \psi) = \oint_{\partial S} d\mathbf{l} \psi \tag{64}$$

to the part of the first term inside the brackets, to get

$$\begin{aligned}
-\epsilon_{ijk} n_j \left[\int_S ds'\, \hat{\mathbf{n}}' \times \nabla' \big(f(\mathbf{x}') \partial_k G^R(\mathbf{x}, \mathbf{x}')\big) \right]_i \\
&= -\epsilon_{ijk} n_j \left[\oint_{\partial S} d\mathbf{l}' f(\mathbf{x}') \partial_k G^R(\mathbf{x}, \mathbf{x}') \right]_i && (65)
\end{aligned}$$

$$= -\epsilon_{ijk} n_j \oint_{\partial S} dl'_i f(\mathbf{x}') \partial_k G^R(\mathbf{x}, \mathbf{x}') \tag{66}$$

$$= -\oint_{\partial S} d\mathbf{l}' \cdot (\hat{\mathbf{n}} \times \nabla G^R(\mathbf{x}, \mathbf{x}')) f(\mathbf{x}'), \tag{67}$$

which is integrable. To evaluate the rest, use the fact that

$$\nabla G^R(\mathbf{x}, \mathbf{x}') = -\nabla' G^R(\mathbf{x}, \mathbf{x}') \tag{68}$$

to write

$$\begin{aligned} &\epsilon_{ijk} n_j \left[\int_S ds' \, \partial_k G^R(\mathbf{x}, \mathbf{x}')(\hat{\mathbf{n}}' \times \nabla' f(\mathbf{x}')) \right]_i \\ &= -\int_S ds' \, \partial'_k G^R(\mathbf{x}, \mathbf{x}') \epsilon_{kij} (\hat{\mathbf{n}}' \times \nabla' f(\mathbf{x}'))_i n_j \qquad (69) \\ &= -\int_S ds' \, \nabla' G^R(\mathbf{x}, \mathbf{x}') \cdot [(\hat{\mathbf{n}}' \times \nabla' f(\mathbf{x}')) \times \hat{\mathbf{n}}]. \qquad (70) \end{aligned}$$

At the field point, the vector in brackets becomes

$$(\hat{\mathbf{n}} \times \nabla' f(\mathbf{x}')) \times \hat{\mathbf{n}} = -\hat{\mathbf{n}} \times (\hat{\mathbf{n}} \times \nabla' f(\mathbf{x}')) = \nabla'_{\parallel} f(\mathbf{x}'). \tag{71}$$

Some notation from differential geometry is useful at this point: $\partial_\mu \mathbf{x} \equiv \partial \mathbf{x}/\partial u^\mu$ is the derivative of the surface with respect to surface parameter u^μ; $g_{\mu\nu}$ is the metric tensor given by $\partial_\mu \mathbf{x} \cdot \partial_\nu \mathbf{x}$; $g^{\mu\nu}$ is the inverse of $g_{\mu\nu}$; g is the determinant of $g_{\mu\nu}$; and $\partial'_\mu f$ represents the derivative of f with respect to u^μ, i.e., $\partial'_\mu f \equiv \partial f(\mathbf{x}'(\mathbf{u}))/\partial u^\mu$.

Thus, in the language of differential geometry, the vector in brackets becomes

$$\partial'^\mu f \partial'_\mu \mathbf{x}' = g^{\mu\nu} \partial'_\nu f \partial'_\mu \mathbf{x}' = \frac{\alpha^\mu \partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}} \tag{72}$$

when α^μ is defined as

$$\sqrt{g(\mathbf{u})} g^{\mu\nu} \partial'_\nu f \tag{73}$$

evaluated at the field point. Therefore, we may write

$$\begin{aligned} &-\int_S ds' \, \nabla' G^R(\mathbf{x}, \mathbf{x}') \cdot [(\hat{\mathbf{n}}' \times \nabla' f(\mathbf{x}')) \times \hat{\mathbf{n}}] \\ &= \int_S ds' \, \nabla' G^R(\mathbf{x}, \mathbf{x}') \cdot \left[\hat{\mathbf{n}} \times (\hat{\mathbf{n}}' \times \nabla' f(\mathbf{x}')) + \frac{\alpha^\mu \partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}} \right] \\ &\quad - \int_S ds' \, \nabla' G^R(\mathbf{x}, \mathbf{x}') \cdot \left[\frac{\alpha^\mu \partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}} \right]. \end{aligned} \tag{74}$$

The first term is integrable because the zero of

$$\left[\hat{\mathbf{n}} \times (\hat{\mathbf{n}}' \times \nabla' f(\mathbf{x}')) + \frac{\alpha^\mu \partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}} \right] \tag{75}$$

at the field point cancels one of the two poles from $\nabla' G^R(\mathbf{x}, \mathbf{x}')$ at the field point. The other term may be rewritten as

$$-\int_S ds' \, \nabla' G^R(\mathbf{x}, \mathbf{x}') \cdot \left[\frac{\alpha^\mu \partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}}\right]$$

$$= -\int_S ds' \, \nabla'_\parallel G^R(\mathbf{x}, \mathbf{x}') \cdot \left[\frac{\alpha^\mu \partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}}\right] \tag{76}$$

$$= -\int_S ds' \, \nabla' \cdot \left(G^R(\mathbf{x}, \mathbf{x}') \frac{\alpha^\mu \partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}}\right),$$

$$+ \alpha^\mu \int_S ds' \, G^R(\mathbf{x}, \mathbf{x}') \nabla'_\parallel \cdot \left[\frac{\partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}}\right], \tag{77}$$

where the last step shows the result of integrating by parts. The part of the first term in parentheses has no normal component so it can be converted to a boundary integral using the divergence theorem for open surfaces (see Appendix B):

$$-\int_S ds' \, \nabla'_\parallel \cdot \left(G^R(\mathbf{x}, \mathbf{x}') \frac{\alpha^\mu \partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}}\right)$$

$$= -\oint_{\partial S} (d\mathbf{l}' \times \hat{\mathbf{n}}') \cdot \left(G^R(\mathbf{x}, \mathbf{x}') \frac{\alpha^\mu \partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}}\right) \tag{78}$$

$$= -\oint_{\partial S} d\mathbf{l}' \cdot \left[\hat{\mathbf{n}}' \times \left(\alpha^\mu \partial'_\mu \mathbf{x}'\right)\right] \frac{G^R(\mathbf{x}, \mathbf{x}')}{\sqrt{g(\mathbf{u})}}. \tag{79}$$

The second term is zero since (see Appendix C)

$$\nabla'_\parallel \cdot \left[\frac{\partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}}\right] = 0. \tag{80}$$

Putting the various terms together, we arrive at the numerically tractable expression for the integral needed to compute local corrections for the hypersingular component of the kernel.

$$\int_S ds' \left(k^2 (\hat{\mathbf{n}} \cdot \hat{\mathbf{n}}') G^R(\mathbf{x}, \mathbf{x}') f(\mathbf{x}') + \nabla' G^R(\mathbf{x}, \mathbf{x}') \cdot \left[\hat{\mathbf{n}} \times (\hat{\mathbf{n}}' \times \nabla' f(\mathbf{x}')) + \frac{\alpha^\mu \partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}}\right]\right)$$

$$- \oint_{\partial S} d\mathbf{l}' \cdot \left((\hat{\mathbf{n}} \times \nabla G^R(\mathbf{x}, \mathbf{x}')) f(\mathbf{x}') + \left(\hat{\mathbf{n}}' \times \left(\alpha^\mu \partial'_\mu \mathbf{x}'\right)\right) \frac{G^R(\mathbf{x}, \mathbf{x}')}{\sqrt{g(\mathbf{u})}}\right), \tag{81}$$

where

$$\alpha^\mu \equiv \sqrt{g(\mathbf{u})} g^{\mu\nu} \partial'_\nu f(\mathbf{x}'(\mathbf{u})), \tag{82}$$

evaluated at the field point. The first integral is a surface integral whose integrand diverges no worse than $1/r$ near the field point; the second is a boundary integral of a regular function (so long as the field point is never situated on the boundary).

A.3. Three-dimensional vector.

A.3.a. $G(r)(\mathbf{t}(\mathbf{x}) \cdot \mathbf{t}'(\mathbf{x}'))$.

This kernel is identical to $G(r)$ in the 3D scalar case, except that the regular function with which it must be convolved is the inner product of a tangent vector $\mathbf{t}(\mathbf{x})$ at the field point and a tangent vector $\mathbf{t}'(\mathbf{x}')$ at the source point. Four sets of local corrections must be computed for each field point since there are two independent tangent vectors at each field point and two at each source point.

A.3.b. $\mathbf{t}(\mathbf{x}) \cdot (\nabla' G(r) \times \mathbf{t}'(\mathbf{x}'))$.

$$\begin{aligned}
&\mathbf{t} \cdot (\nabla' G(r) \times \mathbf{t}'(\mathbf{x}')) \\
&\quad = (ikr - 1)\, e^{ikr} \frac{(\mathbf{t}'(\mathbf{x}') \times \mathbf{t}(\mathbf{x})) \cdot \mathbf{r}}{r^2} \frac{1}{r} \qquad (83)
\end{aligned}$$

$$\begin{aligned}
&\quad = \underbrace{ik^3 \frac{\overbrace{\left(\frac{\sin(kr)}{kr} - \cos(kr)\right)}^{\text{regular}}}{(kr)^2} \overbrace{((\mathbf{t}(\mathbf{x}) \times \mathbf{t}'(\mathbf{x}')) \cdot \mathbf{r})}^{\text{regular}}}_{\text{regular}} \\
&\qquad + \underbrace{\overbrace{(\cos(kr) + (kr)\sin(kr))}^{\text{regular}} \overbrace{\frac{((\mathbf{t}(\mathbf{x}) \times \mathbf{t}'(\mathbf{x}')) \cdot \mathbf{r})}{r^2} \frac{1}{r}}^{\text{singular}}}_{\text{singular}}. \qquad (84)
\end{aligned}$$

The analysis of the singular component is as follows. We can write $\mathbf{t}(\mathbf{x})$ in terms of surface derivatives at the field point

$$\mathbf{t}(\mathbf{x}) = \zeta^{\mu} \partial_{\mu} \mathbf{x} \qquad (85)$$

with some pair of coefficients ζ^{μ}, $\mu = 1, 2$. Letting $\mathbf{u}'$ denote the parameterization of the source point relative to the field point, we can write the expansions for $\mathbf{t}'(\mathbf{x}')$ and $\mathbf{r}(\mathbf{x}')$ about the field point,

$$\mathbf{t}'(\mathbf{x}') = \xi^{\rho} \partial'_{\rho} \mathbf{x}' = \xi^{\rho} \left(\partial_{\rho} \mathbf{x} + \partial_{\rho} \partial_{\sigma} \mathbf{x}\, u'^{\sigma} + \cdots \right), \qquad (86)$$

for some other pair of coefficients ξ^{ρ} with $\rho = 1, 2$ and

$$\mathbf{r}(\mathbf{x}') = \partial_{\tau} \mathbf{x}\, u'^{\tau} + \cdots. \qquad (87)$$

Then

$$\begin{aligned}
((\mathbf{t}(\mathbf{x}) \times \mathbf{t}'(\mathbf{x}')) \cdot \mathbf{r}) &= \zeta^{\mu} \xi^{\rho} \left(\partial_{\mu} \mathbf{x} \times \partial_{\rho} \mathbf{x} + \partial_{\mu} \mathbf{x} \times \partial_{\rho} \partial_{\sigma} \mathbf{x}\, u'^{\sigma} + \cdots \right) \cdot \left(\partial_{\tau} \mathbf{x}\, u'^{\tau} + \cdots \right) \qquad (88) \\
&= \zeta^{\mu} \xi^{\rho} ((\partial_{\mu} \mathbf{x} \times \partial_{\rho} \partial_{\sigma} \mathbf{x}) \cdot \partial_{\tau} \mathbf{x}) u'^{\sigma} u'^{\tau} + \cdots. \qquad (89)
\end{aligned}$$

Since the leading term in $1/r^2$ is also second order in $\mathbf{u}'$, the ratio $((\mathbf{t}(\mathbf{x}) \times \mathbf{t}'(\mathbf{x}')) \cdot \mathbf{r})/r^2$ does not diverge in the limit as $r \to 0$. However, like the factors $(\hat{\mathbf{n}}' \cdot \mathbf{r})/r^2$ and $(\hat{\mathbf{n}} \cdot \mathbf{r})/r^2$ from the 3D scalar case, this ratio is not a regular function unless the principal radii of curvature at the field point are identical. Computation of local correction integrals for each combination of tangent vectors at the field and source points proceeds as in the corresponding 3D scalar case.

A.3.c. $(\mathbf{t}(\mathbf{x}) \cdot \nabla)(\nabla' G(r) \cdot \mathbf{t}'(\mathbf{x}'))$.

$$
\begin{aligned}
&(\mathbf{t}(\mathbf{x}) \cdot \nabla)(\nabla' G(r) \cdot \mathbf{t}'(\mathbf{x}')) \\
&\quad = (\mathbf{t} \cdot \mathbf{t}')\left(\frac{1 - ikr}{r^3}\right)e^{ikr} + (\mathbf{t} \cdot \mathbf{r})(\mathbf{t}' \cdot \mathbf{r})\left(\frac{k^2r^2 + 3ikr - 3}{r^5}\right)e^{ikr} \qquad (90)
\end{aligned}
$$

$$
\begin{aligned}
&\quad = ik^3 \underbrace{\left(\overbrace{\frac{\left(\frac{\sin(kr)}{kr} - \cos(kr)\right)}{(kr)^2}}^{\text{regular}} \overbrace{(\mathbf{t} \cdot \mathbf{t}')}^{\text{regular}} + k^2 \overbrace{\frac{\frac{\sin(kr)}{kr} - 3\left(\frac{\frac{\sin(kr)}{kr} - \cos(kr)}{(kr)^2}\right)}{(kr)^2}}^{\text{regular}} \overbrace{(\mathbf{t} \cdot \mathbf{r})(\mathbf{t}' \cdot \mathbf{r})}^{\text{regular}} \right)}_{\text{regular}} \\
&\qquad + \underbrace{(\mathbf{t} \cdot \nabla)(\nabla' G^R(r) \cdot \mathbf{t}')}_{\text{hypersingular}}. \qquad (91)
\end{aligned}
$$

The result is very similar to that in the 3D scalar case. The real part of $G(r)$, namely $G^R(r) \equiv \cos(kr)/r$, produces a hypersingular term that is not (in general) integrable because it diverges like $1/r^3$ relative to the field point. We now show how to manipulate it into a form that can be evaluated numerically when the region of integration contains the field point.

Reformulating the integral of the hypersingular term begins with an integration by parts:

$$
\begin{aligned}
&\int_S ds'(\mathbf{t}(\mathbf{x}) \cdot \nabla)(\nabla' G^R(\mathbf{x}, \mathbf{x}') \cdot \mathbf{t}'(\mathbf{x}')) \\
&\quad = \int_S ds'\, \mathbf{t}'(\mathbf{x}') \cdot \nabla'_{\parallel}(\mathbf{t}(\mathbf{x}) \cdot \nabla G^R(\mathbf{x}, \mathbf{x}')) \qquad (92) \\
&\quad = \int_S ds'\, \nabla'_{\parallel} \cdot [\mathbf{t}'(\mathbf{x}')(\mathbf{t}(\mathbf{x}) \cdot \nabla G^R(\mathbf{x}, \mathbf{x}'))] \\
&\qquad - \int_S ds'(\mathbf{t}(\mathbf{x}) \cdot \nabla G^R(\mathbf{x}, \mathbf{x}'))(\nabla'_{\parallel} \cdot \mathbf{t}'(\mathbf{x}')). \qquad (93)
\end{aligned}
$$

The first term on the last line can be converted to a boundary integral using the divergence theorem for open surfaces (see Appendix B) and the fact that the argument of $\nabla'_{\parallel}\cdot$ is tangential to the surface:

$$
\int_S ds'\, \nabla'_{\parallel} \cdot [\mathbf{t}'(\mathbf{x}')(\mathbf{t}(\mathbf{x}) \cdot \nabla G^R(\mathbf{x}, \mathbf{x}'))] = \oint_{\partial S} dl(\hat{\mathbf{e}}' \cdot \mathbf{t}'(\mathbf{x}'))(\mathbf{t}(\mathbf{x}) \cdot \nabla G^R(\mathbf{x}, \mathbf{x}')). \qquad (94)
$$

The second term is

$$
-\int_S ds'(\mathbf{t}(\mathbf{x}) \cdot \nabla G^R(\mathbf{x}, \mathbf{x}'))(\nabla'_{\parallel} \cdot \mathbf{t}'(\mathbf{x}')) = \int_S ds'\, \nabla' G^R(\mathbf{x}, \mathbf{x}') \cdot [\mathbf{t}(\mathbf{x})(\nabla'_{\parallel} \cdot \mathbf{t}'(\mathbf{x}'))]. \qquad (95)
$$

Write this as

$$
\int_S ds'\, \nabla' G^R(\mathbf{x}, \mathbf{x}') \cdot \left[\mathbf{t}(\mathbf{x})(\nabla'_{\parallel} \cdot \mathbf{t}'(\mathbf{x}')) - \frac{\alpha^{\mu}\partial'_{\mu}\mathbf{x}'}{\sqrt{g(\mathbf{u})}}\right] + \int_S ds'\, \nabla' G^R(\mathbf{x}, \mathbf{x}') \cdot \left[\frac{\alpha^{\mu}\partial'_{\mu}\mathbf{x}'}{\sqrt{g(\mathbf{u})}}\right]. \qquad (96)
$$

where the constant α^μ is chosen to make $\mathbf{t}(\mathbf{x})(\nabla' \cdot \mathbf{t}'(\mathbf{x}'))$ and $\alpha^\mu \partial'_\mu \mathbf{x}'/\sqrt{g(\mathbf{u})}$ equal at the field point. In other words, α^μ is defined as

$$\sqrt{g(\mathbf{u})}g^{\mu\nu}(\mathbf{t}(\mathbf{x}) \cdot \partial'_\nu \mathbf{x}')(\nabla' \cdot \mathbf{t}'(\mathbf{x}')) \tag{97}$$

evaluated at the field point. The first term is integrable because the zero of

$$\left[\mathbf{t}(\mathbf{x})(\nabla'_\parallel \cdot \mathbf{t}'(\mathbf{x}')) - \frac{\alpha^\mu \partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}}\right] \tag{98}$$

at the field point cancels one of the two poles from $\nabla' G^R(\mathbf{x}, \mathbf{x}')$ at the field point. As shown in the 3D scalar case, the second term reduces to the boundary integral:

$$\begin{aligned} \int_S ds' \, \nabla' G^R(\mathbf{x}, \mathbf{x}') \cdot \left[\frac{\alpha^\mu \partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}}\right] &= \oint_{\partial S} d\mathbf{l}' \cdot \left[\hat{\mathbf{n}}' \times \left(\alpha^\mu \partial'_\mu \mathbf{x}'\right)\right] \frac{G^R(\mathbf{x}, \mathbf{x}')}{\sqrt{g(\mathbf{u})}} & (99) \\ &= \oint_{\partial S} dl' \, \hat{\mathbf{e}}' \cdot \left(G^R(\mathbf{x}, \mathbf{x}') \frac{\alpha^\mu \partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}}\right). & (100) \end{aligned}$$

Putting the various terms together, we arrive at the numerically tractable expression for the integral needed to compute local corrections for the hypersingular component of the kernel,

$$\begin{aligned} &\int_S ds' \, \nabla' G^R(\mathbf{x}, \mathbf{x}') \cdot \left[\mathbf{t}(\mathbf{x})(\nabla'_\parallel \cdot \mathbf{t}'(\mathbf{x}')) - \frac{\alpha^\mu \partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}}\right] \\ &\quad + \oint_{\partial S} dl' \, \hat{\mathbf{e}}' \cdot \left((\mathbf{t}(\mathbf{x}) \cdot \nabla G^R(\mathbf{x}, \mathbf{x}'))\mathbf{t}'(\mathbf{x}') + G^R(\mathbf{x}, \mathbf{x}') \frac{\alpha^\mu \partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}}\right), \end{aligned} \tag{101}$$

where

$$\alpha^\mu \equiv \sqrt{g(\mathbf{u})}g^{\mu\nu}(\mathbf{t}(\mathbf{x}) \cdot \partial'_\nu \mathbf{x}')(\nabla'_\parallel \cdot \mathbf{t}'(\mathbf{x}')) = \sqrt{g(\mathbf{u})}g^{\mu\nu}(\mathbf{t}(\mathbf{x}) \cdot \partial'_\nu \mathbf{x}')\left(g^{\rho\sigma} \partial'_\rho \mathbf{t}' \cdot \partial'_\sigma \mathbf{x}'\right). \tag{102}$$

evaluated at the field point. The first integral is a surface integral whose integrand diverges no worse than $1/r$; the second is a boundary integral of a regular function (so long as the field point is never situated on the boundary).

If, as suggested in Section III.C.3, the μth tangent vector at the field point (with surface parameter $\mathbf{u}_0$) is given by

$$\mathbf{t}_\mu(\mathbf{u}) = \partial_\mu \mathbf{x}(\mathbf{u}) \tag{103}$$

and the νth vector testing function associated with scalar testing function $f^{(k)}(\mathbf{u})$ is given by

$$\mathbf{t}^{(k)}_\nu(\mathbf{u}) = \frac{\partial_\nu \mathbf{x}(\mathbf{u})}{\sqrt{g(\mathbf{u})}} f^{(k)}(\mathbf{u}). \tag{104}$$

then Eq. (101) simplifies to

$$\int_S ds'\, \nabla' G^R(\mathbf{x},\mathbf{x}') \cdot \left(\partial_\mu \mathbf{x} \partial'_\nu f^{(k)}(\mathbf{u}) - \partial'_\mu \mathbf{x}' \partial'_\nu f^{(k)}(\mathbf{u}_0)\right) / \sqrt{g(\mathbf{u})}$$
$$+ \oint_{\partial S} dl'\, \hat{\mathbf{e}}' \cdot \left(G^R(\mathbf{x},\mathbf{x}') \partial'_\nu f^{(k)}(\mathbf{u}_0) \partial'_\mu \mathbf{x}' + \left(\partial_\mu \mathbf{x} \cdot \nabla G^R(\mathbf{x},\mathbf{x}')\right) f^{(k)}(\mathbf{u}) \partial'_\nu \mathbf{x}'\right) / \sqrt{g(\mathbf{u})}. \tag{105}$$

B. *Divergence Theorem for Open Surfaces*

Substitute

$$\mathbf{B} = \hat{\mathbf{n}} \times \mathbf{A} \tag{106}$$

into Stokes's theorem

$$\int_S ds\, \hat{\mathbf{n}} \cdot (\nabla \times \mathbf{B}) = \oint_{\partial S} d\mathbf{l} \cdot \mathbf{B} \tag{107}$$

to get

$$\int_S ds\, \hat{\mathbf{n}} \cdot (\nabla \times (\hat{\mathbf{n}} \times \mathbf{A}))$$
$$= \int_S ds\, \hat{\mathbf{n}} \cdot [\hat{\mathbf{n}}(\nabla \cdot \mathbf{A}) - (\hat{\mathbf{n}} \cdot \nabla)\mathbf{A} - \mathbf{A}(\nabla \cdot \hat{\mathbf{n}}) + (\mathbf{A} \cdot \nabla)\hat{\mathbf{n}}] \tag{108}$$
$$= \int_S ds[(\nabla_\parallel \cdot \mathbf{A}) - (\hat{\mathbf{n}} \cdot \mathbf{A})(\nabla \cdot \hat{\mathbf{n}})] \tag{109}$$
$$= \oint_{\partial S} d\mathbf{l} \cdot (\hat{\mathbf{n}} \times \mathbf{A}) \tag{110}$$
$$= \oint_{\partial S} (d\mathbf{l} \times \hat{\mathbf{n}}) \cdot \mathbf{A} \tag{111}$$
$$= \oint_{\partial S} dl\, \hat{\mathbf{e}} \cdot \mathbf{A}, \tag{112}$$

where we have used the definition of tangential gradient

$$\nabla_\parallel \equiv \nabla - \hat{\mathbf{n}}(\hat{\mathbf{n}} \cdot \nabla) \tag{113}$$

and the following equation which relates the vector line element $d\mathbf{l}$ and the surface normal $\hat{\mathbf{n}}$ to the scalar line element dl and the unit edge vector $\hat{\mathbf{e}}$,

$$d\mathbf{l} \times \hat{\mathbf{n}} = dl\, \hat{\mathbf{e}}, \tag{114}$$

and the observation that

$$\hat{\mathbf{n}} \cdot [(\mathbf{A} \cdot \nabla)\hat{\mathbf{n}}] = [(\mathbf{A} \cdot \nabla)\hat{\mathbf{n}}] \cdot \hat{\mathbf{n}} = \frac{1}{2}(\mathbf{A} \cdot \nabla)(\hat{\mathbf{n}} \cdot \hat{\mathbf{n}}) = 0. \tag{115}$$

In other words, the divergence theorem for open surfaces is

$$\int_S ds[(\nabla_\parallel \cdot \mathbf{A}) - (\hat{\mathbf{n}} \cdot \mathbf{A})(\nabla \cdot \hat{\mathbf{n}})] = \oint_{\partial S} dl\, \hat{\mathbf{e}} \cdot \mathbf{A} = \oint_{\partial S} (d\mathbf{l} \times \hat{\mathbf{n}}) \cdot \mathbf{A}. \tag{116}$$

which simplifies to

$$\int_S ds(\nabla_{\parallel} \cdot \mathbf{A}) = \oint_{\partial S} dl\, \hat{\mathbf{e}} \cdot \mathbf{A} = \oint_{\partial S} (d\mathbf{l} \times \hat{\mathbf{n}}) \cdot \mathbf{A} \tag{117}$$

when **A** is everywhere tangential to S.

C. *Proof that* $\nabla'_{\parallel} \cdot [\partial'_\mu \mathbf{x}'/\sqrt{g(\mathbf{u})}] = 0$

Note. Summation over repeated indices is implied:

$$\begin{aligned}
\nabla'_{\parallel} \cdot \left[\frac{\partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}}\right] &= \partial'^{\rho}\left(\frac{\partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}}\right) \cdot \partial'_\rho \mathbf{x}' \\
&= g^{\rho\sigma}\partial'_\sigma\left(\frac{\partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}}\right) \cdot \partial'_\rho \mathbf{x}' \\
&= g^{\rho\sigma}\left(\frac{\partial'_\sigma \partial'_\mu \mathbf{x}'}{\sqrt{g(\mathbf{u})}} - \frac{\partial'_\mu \mathbf{x}'}{2\sqrt{g(\mathbf{u})^3}}\partial'_\sigma g(\mathbf{u})\right) \cdot \partial'_\rho \mathbf{x}' \\
&= \frac{g^{\rho\sigma}}{\sqrt{g(\mathbf{u})}}\left(\partial'_\sigma \partial'_\mu \mathbf{x}' \cdot \partial'_\rho \mathbf{x}' - \frac{\partial'_\mu \mathbf{x}' \cdot \partial'_\rho \mathbf{x}'}{2g(\mathbf{u})}(2g(\mathbf{u})g^{\alpha\beta}\partial'_\alpha \mathbf{x}' \cdot \partial'_\sigma \partial'_\beta \mathbf{x}')\right) \\
&= \frac{g^{\rho\sigma}}{\sqrt{g(\mathbf{u})}}(\partial'_\rho \mathbf{x}' \cdot \partial'_\sigma \partial'_\mu \mathbf{x}' - g_{\mu\rho} g^{\alpha\beta}\partial'_\alpha \mathbf{x}' \cdot \partial'_\sigma \partial'_\beta \mathbf{x}') \\
&= \frac{1}{\sqrt{g(\mathbf{u})}}(g^{\rho\sigma}\partial'_\rho \mathbf{x}' \cdot \partial'_\sigma \partial'_\mu \mathbf{x}' - \delta^\sigma_\mu g^{\alpha\beta}\partial'_\alpha \mathbf{x}' \cdot \partial'_\sigma \partial'_\beta \mathbf{x}') \\
&= \frac{1}{\sqrt{g(\mathbf{u})}}(g^{\rho\sigma}\partial'_\rho \mathbf{x}' \cdot \partial'_\sigma \partial'_\mu \mathbf{x}' - g^{\alpha\beta}\partial'_\alpha \mathbf{x}' \cdot \partial'_\mu \partial'_\beta \mathbf{x}') \\
&= \frac{1}{\sqrt{g(\mathbf{u})}}(g^{\rho\sigma}\partial'_\rho \mathbf{x}' \cdot \partial'_\sigma \partial'_\mu \mathbf{x}' - g^{\rho\sigma}\partial'_\rho \mathbf{x}' \cdot \partial'_\sigma \partial'_\mu \mathbf{x}') = 0.
\end{aligned}$$

ACKNOWLEDGMENTS

We are grateful to Drs. Vladimir Rokhlin and Leslie Greengard for considerable guidance regarding the use of high-order Nyström discretizations in scattering calculations. We also thank Dr. George Valley for reviewing the manuscript and offering useful suggestions for improvement. The U.S. Government's right to retain a nonexclusive royalty-free license in and to the copyright covering this paper, for governmental purposes, is acknowledged.

REFERENCES

1. J. J. Ottusch, Performance comparison of FastScat(TM) and RAM2D, in *Presentations of Electromagnetic Code Consortium Annual Meeting, Albuquerque, NM, May 1994*.
2. L. R. Hamilton, J. J. Ottusch, M. A. Stalzer, R. S. Turley, J. L. Visher, and S. M. Wandzura, FastScat benchmark data, in *Proc. 1994 HAVE FORUM Symposium, Wright Patterson AFB, OH 454-7523*, Vol. I, p. 255 (Wright Laboratory, Feb. 1995). [WL-TR-95-6003]
3. S. Wandzura, High-order discretization of integral equations with singular kernels, in *IEEE Antennas Propag. Soc. Int. Sympos. Digest, Newport Beach, CA*, Vol. 1, p. 792 (IEEE, New York, June 1995).
4. J. S. Kot, Computer modelling of mm-wave integrated circuit antennas using the Nyström method, in *International Conference on Computation in Electromagnetics*, Vol. 3, p. 25 (IEEE Press, New York, Nov. 1991).

5. R. Kress, Numerical solution of boundary integral equations in time-harmonic electromagnetic scattering, *Electromagnetics* **10**, 1 (1990).
6. W. H. Press, B. P. Flannery, S. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C—The Art of Scientific Computing* (Cambridge Univ. Press, Cambridge, 1988).
7. L. M. Delves and J. L. Mohamed, *Computational Methods for Integral Equations* (Cambridge Univ. Press, New York, 1985).
8. J. Strain, Locally-corrected multidimensional quadrature rules for singular functions, *SIAM J. Sci. Comput.* **16**, 992 (1995).
9. S. M. Rao, D. R. Wilton, and A. W. Glisson, Electromagnetic scattering by surfaces of arbitrary shape, *IEEE Trans. Antennas Propag.* **AP-30**, 409 (1982).
10. S. M. Wandzura, Electric current basis functions for curved surfaces, *Electromagnetics* **12**, 77 (1992).
11. R. Coifman, V. Rokhlin, and S. Wandzura, The fast multipole method: A pedestrian prescription, *IEEE Antennas Propag. Soc. Mag.* **35**, 7 (1993).
12. E. Bleszynski, M. Bleszynski, and T. Jaroszewicz, AIM: adaptive integral method for solving large-scale electromagnetic scattering and radiation problems, *Radio Sci.* **31**, 1225 (1996).
13. V. Rokhlin and M. A. Stalzer, Scalability of the fast multipole method for the Helmholtz equation, in *Eighth SIAM Conference on Parallel Processing for Scientific Computing, Minneapolis, MN* (SIAM, Philadelphia, 1997).
14. J. M. Song and W. C. Chew, Multilevel fast-multipole algorithm for solving combined field equations of electromagnetic scattering, *Microwave Opt. Technol. Lett.* **10**, 14 (1995).
15. S. D. Gedney, J. J. Ottusch, P. Petre, J. Visher, and S. Wandzura, Efficient high-order discretization schemes for integral equation methods, in *IEEE Symposium on Antennas and Propagation, Montreal, Canada, July 1997.*
16. J. N. Lyness and D. Jespersen, Moderate degree symmetric quadrature rules for the triangle, *J. Inst. Math. Appl.* **15**, 19 (1975).
17. N. Morita, N. Kumagai, and J. R. Mautz, *Integral Equation Methods for Electromagnetics* (Artech, Boston, 1990).
18. D. Colton and R. Kress, *Integral Equation Methods in Scattering Theory* (Wiley, New York, 1983).
19. J. J. Bowman, T. B. A. Senior, and P. L. E. Uslenghi (Eds.), *Electromagnetic and Acoustic Scattering by Simple Shapes* (Hemisphere, New York, 1987).
20. L. Canino, L. Hamilton, J. J. Ottusch, R. Ross, J. Visher, and S. Wandzura, FastScat performance on EMCC benchmark cases, in *Presentations of Electromagnetic Code Consortium Annual Meeting, Rome, NY, May 1996.*
21. *User's Manual for FISC (Fast Illinois Solver Code)* (Center for Computational Electromagnetics at the University of Illinois and DEMACO, Inc., 1997).
22. V. Rokhlin, personal communication, 1997.
23. A. W. Maue, Toward formulation of a general diffraction problem via an integral equation, *Z. Phys.* **126**, 601 (1949).
24. J.-H. Ma, V. Rokhlin, and S. Wandzura, Generalized Gaussian quadrature rules for systems of arbitrary functions, *SIAM J. Numer. Anal.* **33**, 971 (1996).
25. M. G. Duffy, Quadrature over a pyramid or cube of integrands with a singularity at a vertex, *J. Numer. Anal.* **19**, 1260 (1982).

A Prescription for the Multilevel Helmholtz FMM

Mark F. Gyure and Mark A. Stalzer
HRL Laboratories

◆ ◆ ◆

The authors describe a multilevel Helmholtz FMM as a way to compute the field caused by a collection of source points at an arbitrary set of field points. Their description focuses on the algorithm's mathematical basics, so that it can be applied to a variety of applications.

◆

The *fast multipole method* for the scalar Helmholtz equation, $(\nabla^2 + k^2)\Psi = 0$, is commonly used to compute acoustic- and electromagnetic-scattering cross sections.[1,2] Ronald Coifman, Vladimir Rokhlin, and Stephen Wandzura[3] described a single-level scheme, which has been implemented in two and three dimensions for scalar and vector scattering problems.[4–6] The method has been subsequently extended to multiple levels, again with an emphasis on electromagnetic scattering.[7,8]

In this article, we'll focus on the basic multilevel FMM algorithm as a way to quickly compute the field caused by a collection of Helmholtz source points at an arbitrary field point. To keep our description of the implementation simple, we'll assume that the field is desired at each source point, as would normally be the case when constructing an impedance matrix for a physical problem. Through this basic, but detailed, description, we hope to make the multilevel Helmholtz FMM more accessible for a variety of problems.

The mathematical preliminaries

Previous research on the FMM has taken two approaches. The first[3] starts from the standard integral equation for a field arising from an arbitrary source distribution assumed to be localized to surfaces:

$$\phi(r) = \int_S d\mathbf{r}' \frac{e^{ik|\mathbf{r}-\mathbf{r}'|}}{|\mathbf{r}-\mathbf{r}'|}\sigma(\mathbf{r}') . \qquad (1)$$

This approach then manipulates the integral equation by substituting two identities: one, a form of the Gegenbauer addition theorem, and the other, a plane wave expansion for spherical Bessel functions. The result is an expression for Equation 1, from which it is straightforward to construct an algorithm for computing the field in $O(N^{3/2})$ operations, where N is the number of unknowns describing the entire source distribution. Extension to a multilevel FMM that scales as $O(N \log^2 N)$ is also possible through this approach.

The other approach, taken by Rokhlin in the original Helmholtz FMM paper[2] and the one we use here, uses the language of multipole expansions that are valid exterior or interior to groups containing an arbitrary number of source or field points. In this approach, the essential point is that diagonal transforms exist for translating the origins of both interior and exterior expansions of charge distributions as well as for converting exterior expansions to interior expansions.

Rokhlin has already described the mathematical details involved in constructing exterior and interior expansions, and has provided proofs of the various theorems involving translation operators.[2] We'll now provide a concep-

1070-9924/98/$10.00 © 1998 IEEE

tual framework in which manipulation of off-centered expansions through diagonal translation operators and efficient transforms is a completely natural way to view the FMM. This approach is not only more general, but also better-suited for describing the multilevel FMM. In particular, it also makes clear why the interpolation and filtering steps that are necessary in the multilevel FMM must be treated carefully.

Multipole expansions and translation operators

Consider two well-separated spheres of radius R_1 and R_2, each containing a collection of points. We'll take the points inside R_1 to be Helmholtz point sources and the points inside R_2 to be field points at which we would like to evaluate the field caused by the collection of sources in R_1. This field, written as a multipole expansion[2] valid outside R_1, is

$$\psi(\mathrm{r}) = \sum_{lm} \beta_{lm} h_l(kr) Y_{lm}(\theta,\phi), \tag{2}$$

where r, θ, and ϕ are relative to a coordinate system centered in R_1, $h_l(kr)$ are spherical Hankel functions of the first kind, and $Y_{lm}(\theta, \phi)$ are the normalized spherical harmonics. We'll refer to this expansion as an exterior or h-expansion. Similarly, we can write an expression for the field valid inside R_2:

$$\phi(\mathrm{r}) = \sum_{lm} \alpha_{lm} j_l(kr) Y_{lm}(\theta,\phi), \tag{3}$$

where r, θ, and ϕ are now relative to a coordinate system centered in R_2, and $j_l(kr)$ are spherical Bessel functions. We'll refer to this expansion as an interior or j-expansion. For the moment, we will consider both of these to be infinite sums. The FMM then rests on three observations:

- The origin of the h-expansion (Equation 2) can be shifted arbitrarily inside R_1, and a new set of coefficients, $\bar{\beta}_{lm}$, can be computed for this new expansion. The same holds for shifting a j-expansion (see Equation 3) arbitrarily to a new origin inside R_2, which results in a new set of coefficients, $\bar{\alpha}_{lm}$.
- An h-expansion valid outside R_1 can be translated and converted into a j-expansion valid inside R_2, resulting in a new set of coefficients for the j-expansion, γ_{lm}.
- Most crucial, these translations can be done efficiently by transforming the coefficients into a basis in which both translation operators are diagonal. We'll illustrate this below by constructing a diagonal form for the h-expansion translation operator. The FMM, with one or multiple levels, is now basically a sequence of combinations and translations of multipole coefficients resulting in an expansion for the field that can be easily evaluated at any point inside another group.

Generalized addition theorems for partial wave expansions and their corresponding expressions for the translation of multipole coefficients have been known for many years.[9,10] Rokhlin, however, was the first to realize that these translation operators could accelerate the numerical computation of fields obeying the Helmholtz equation. A general expression exists for translating the coefficients of multipole expansions that are solutions to the Helmholtz equation; the specific forms of interest here are

$$\bar{\beta}_{lm} = \sum_{l'm'} \beta_{l'm'} \sum_{pq} c(lm|l'm'|pq)\lambda_{pq}, \tag{4}$$

$$\bar{\alpha}_{lm} = \sum_{l'm'} \alpha_{l'm'} \sum_{pq} c(lm|l'm'|pq)\lambda_{pq}, \text{ and} \tag{5}$$

$$\gamma_{lm} = \sum_{l'm'} \beta_{l'm'} \sum_{pq} c(lm|l'm'|pq)\mu_{pq} \ . \tag{6}$$

where $c(lm|l'm'|pq)$ is proportional to the well-known 3j symbols involving products of three spherical harmonics:

$$c(lm|l'm'|pq) = i^{l'+\mu-1} \int d\hat{k} Y^*_{lm}(k_\theta, k_\phi) \\ Y_{l'm'}(k_\theta, k_\phi) Y_{\mu\nu}(k_\theta, k_\phi). \tag{7}$$

Following Rokhlin, we will refer to the functions λ_{pq} and μ_{pq} as translation operators. They have the forms

$$\lambda_{pq} = 4\pi j_p(kx_{12}) Y^*_{pq}(\theta_{12}, \phi_{12}) \text{ and} \tag{8}$$

$$\mu_{pq} = 4\pi h_p(kx_{12}) Y^*_{pq}(\theta_{12}, \phi_{12}). \tag{9}$$

In the above expressions, x_{12}, θ_{12}, and ϕ_{12} refer to the coordinates associated with the vector pointing from the expansion's original center to the new center.

The problem with using the above expressions directly in a computational scheme is that an individual coefficient such as $\bar{\beta}_{lm}$ depends on a sum over all the original coefficients $\beta_{l'm'}$ and on a sum over a set of indices associated with the translation operator and 3j symbols. Even with truncation of the multipole expansion to a finite number of terms, L, this approach is not practical. A computationally viable scheme—that is, one that scales no worse than $O(L^2)$—requires diagonalizing this transformation, meaning that each coefficient can be translated independently of all the others. The problem, then, is to find a representation in which this translation is diagonal. This representation is often called the *far-field representation*, and the transform that diagonalizes the two translation operators is the *far-field transform*.

Following Rokhlin, we define the far-field transform and inverse transform of an arbitrary function f as

$$f(k_\theta, k_\phi) = \sum_{lm} i^l Y_{lm}(k_\theta, k_\phi) f_{lm} \text{ and} \tag{10}$$

$$f_{lm} = \int d\hat{k} i^{-l} Y^*_{lm}(k_\theta, k_\phi) f(k_\theta, k_\phi) \quad . \tag{11}$$

This is basically just a spherical harmonic transform that rotates a function from one basis to another in exact analogy to a Fourier transform.

Consider the specific case of translating an h-expansion to a new origin, which means transforming the set of coefficients α_{lm}. By taking the (inverse) far-field transform of α and λ in Equation 5, the far-field transform completely diagonalizes the transformation of the αs—that is,

$$\tilde{\alpha}_{lm} = \int d\hat{k} i^{-l} Y^*_{lm}(k_\theta, k_\phi) \lambda(k_\theta, k_\phi) \alpha(k_\theta, k_\phi) \tag{12}$$

or, equivalently through a far-field transform of Equation 12,

$$\tilde{\alpha}(k_\theta, k_\phi) = \lambda(k_\theta, k_\phi) \alpha(k_\theta, k_\phi) . \tag{13}$$

Even more useful computationally is that the inverse transform λ_{lm} simplifies to

$$\begin{aligned} &\lambda(k_\theta, k_\phi) \\ &= \sum_{lm} i^l Y_{lm}(k_\theta, k_\phi) 4\pi j_l(kx_{12}) Y^*_{lm}(\theta_{12}, \phi_{12}) \\ &= e^{ikx_{12}\cos\gamma} \end{aligned} \quad , \tag{14}$$

where γ is the angle between (θ_{12}, ϕ_{12}) and (k_θ, k_ϕ). Because λ is also the translation operator for j-expansions, the same analysis applies to the translation of interior expansions.

The translation operator λ represents a "local" shift in the group center, retaining the exterior or interior expansion. The translation of an h-expansion into a j-expansion is through the translation operator μ, which, in the far-field basis, has a similar form to λ,

$$\mu(k_\theta, k_\phi) = \sum_l i^l (2l+1) h_l(kx_{12}) P_l(\cos\gamma) , \tag{15}$$

but with considerably different mathematical behavior. The translation operator μ is qualitatively different than λ in that no simpler expression exists. In fact, the infinite sum diverges, and the mathematical consequences of this divergence require careful attention in a rigorous treatment of the FMM. But, a numerical implementation that uses truncated multipole expansions needs only a finite number of terms to achieve a given accuracy in the translation.[2] Hence, the divergence of the infinite sum has no practical consequences.

So far, our description of multipole expansions and translation operators has not covered two significant issues. We haven't discussed any of the theorems that prove that the multipole expansions themselves converge to a specified accuracy in a number of terms approximately proportional to the group radius. Also, we haven't discussed truncation of the series for the h-to-j translation operator, μ. These issues are important in numerical implementation because the algorithm's accuracy depends critically on the number of terms kept in these series. However, Rokhlin has already adequately addressed these issues.[2]

The above expressions for translation operators, together with the far-field transform, are the basic tools used to construct a multilevel FMM algorithm. Clearly, the field caused by a collection of sources inside an arbitrary group G_1 can be evaluated at any point inside a second group G_2 by converting the exterior h-expansion, valid outside G_1, to an interior j-expansion valid inside G_2. We can translate the coefficients of the j-expansion to any point inside G_2. Also, we can calculate the field at that point caused by the sources in G_1 by computing $\tilde{\alpha}_{00}$, the leading term in the j-expansion. No other terms contribute, because the expansion is already centered at the field point where $r = 0$ and all the terms $j_{lm}(0)$ are zero except j_{00}, which is one. Thus, we can evaluate the field directly through the far-field transform as

$$\phi(0) = \tilde{\alpha}_{00} = \frac{1}{\sqrt{4\pi}} \int d\hat{k} \tilde{\alpha}(k_\theta, k_\phi) . \tag{16}$$

Interpolation and filtering

One crucial issue remains in constructing an efficient multilevel algorithm that scales properly. The multilevel Helmholtz FMM works fundamentally the same as the Laplace FMM in that it combines expansions valid inside the original groups to form expansions valid inside correspondingly larger groups with a bigger group radius. This recursive regrouping results in a tree-like structure that has groups of different sizes at different levels of the tree. h- or j-expansions valid for groups at one level must be combined to form expansions valid for either larger or smaller groups at a different level. More specifically, h-expansions from neighboring groups are translated and combined into a single h-expansion representing a larger group when going up the tree, and j-expansions in a large group are translated to smaller groups going down the tree. Let's look at these two operations in more detail.

When combining smaller groups into a larger group, the number of coefficients in the h-expansions representing each of the smaller groups must increase to preserve the accuracy of the source expansion after the coefficients are translated and combined at the new (larger) group center. This is a consequence of translating the h-expansions to origins that are further away than what was allowed by the number of terms in the original expansions. In terms of multipole coefficients, this operation is handled by adding higher-order coefficients, initially zero, and then translating the expansion. The translation mixes the multipole coefficients so that the higher modes are nonzero after the translation. This new expansion can be combined with others being shifted to the same group center by simply adding their coefficients term by term. The problem with implementing this procedure is that the translation operator must be applied in the diagonal far-field representation, not the multipole coefficient representation, for the reasons we described in the previous section. In the far-field basis, the addition of higher-order multipole terms that are zero amounts to an interpolation of the function $\beta(k_\theta, k_\phi)$ onto a denser set of far-field directions (k_θ', k_ϕ'). This interpolation must not introduce spurious high-order multipole terms; otherwise, the algorithm's accuracy is quickly compromised.

A similar problem exists when translating the j-expansions of larger groups to the centers of smaller groups, a procedure that is required when going down the tree. Because a smaller number of multipole terms are needed to represent the field inside a smaller group, the number of terms in the multipole expansion can be decreased with no loss of accuracy. In the far-field representation, this procedure amounts to filtering the function $\alpha(k_\theta, k_\phi)$ to a less dense set of far-field directions (k_θ, k_ϕ). But, just as in the interpolation step described above, the filtering operation must remove only the higher-order multipole coefficients; otherwise, the accuracy is similarly compromised.

The implementation of fast, efficient interpolation or filtering operations is straightforward in principle. Because the translation operators are diagonal in the far-field basis, all FMM implementations keep the h- and j-expansions exclusively in the far-field representation. The interpolation and filtering steps, however, are rigorously defined only in a multipole coefficient basis.

Consider interpolating an h-expansion given by a set of coefficients in the far-field representation $\beta(\theta, \phi)$. The multipole coefficients are given by this far-field transform:

$$\begin{aligned}\beta_{lm} &= \int d(\cos k_\theta) P_l^m(\cos k_\theta) \int d\hat{k} e^{-imk_\phi} \beta(k_\theta, k_\phi) \\ &= \int d(\cos k_\theta) P_l^m(\cos k_\theta) \beta_m(k_\theta) \cdot \end{aligned} \tag{17}$$

We have left out phase and normalization factors in Equation 17. Because filtering or interpolation always involves a transform-inverse pair, we consider these factors as being absorbed into the definition of the multipole coefficients. Assuming a uniform distribution of points in the k_ϕ direction on the unit sphere, a fast Fourier transform (FFT) can easily and efficiently compute $\beta_m(k_\theta)$.

Numerical quadrature handles the remaining part of the transform:

$$\beta_{lm} = \sum_{n=1}^{N} w_n P_l^m(\cos k_{\theta_n}) \beta_m(k_{\theta_n}), \tag{18}$$

where w_n and $k_{\theta n}$ are sets of weights and abscissas for an appropriately defined quadrature rule. Interpolation onto the denser set of points is then handled by the inverse transform

$$\beta'(k_\theta', k_\phi') = \sum_{m=-L'}^{L'} e^{imk_\phi'} \sum_{l=0}^{L'} \beta_{lm} P_l^m(\cos k_\theta') \ , \tag{19}$$

where $L' > L$, the far-field directions (k_θ', k_ϕ') are now a correspondingly denser set of points on the unit sphere, and all the β_{lm} corresponding to

$l > L$ are zero. The filtering step happens in exactly the same way, except that $L' < L$ and the coefficients corresponding to $l > L'$ are simply truncated. In both cases, an FFT can handle the sum on m straightforwardly.

The filtering and interpolating steps will quickly create a serious computational bottleneck and break the scaling of the entire algorithm if not treated properly. Indeed, the primary obstacle to constructing a practical multilevel FMM has been proper handling of these steps.

For this problem, the best solution—desirable because it is exact—is to construct a fast associated Legendre transform.[11,12] When combined with an FFT of the ϕ directions on the unit sphere, this approach results in an operation count that scales no worse than $O(L^2 \log L)$, where L is the order of the spherical harmonic expansion at a given level. This method works in principle but suffers from a large crossover point compared to the "semifast" transform, which also uses the FFT in the k_ϕ direction but uses a slow transform in the k_θ direction, and which scales as L^3. Unfortunately, this crossover point is squarely in the region encountered by problems of large but practical size. Recent work has improved this crossover point somewhat,[12] and we are using the improved algorithm for higher levels of the multilevel FMM, which we'll describe next.

Implementation

Our multilevel FMM implementation consists of two main routines: `setup` and `apply`. `Setup` produces a tree or hierarchy of groups that partition the sources. It uses this tree to precompute the translation operators and other quantities. Using information computed by `setup`, `apply` forms $Z \cdot I$, the value of the field at every source caused by all other sources.

Setup

To construct the tree, `setup` performs the grouping on a cubic lattice where each box edge has the length $D/\sqrt{3}$ (see Figure 1). The group diameter D is picked to minimize the overall operation count and typically ranges from 0.5 to 1.5 wavelengths. At the lowest level (level 0), the routine assigns each elementary source to the box with the closest center. With this base grouping, the grouping process moves on to subsequent levels. At each level l, the size of the boxes doubles, so each box contains up to eight active subboxes. However, because a surface is generally being discretized, the number of active subboxes is usually closer to four. This grouping process continues until all the sources fit in one box. The quantity H is set to the number of levels or height of the tree, and the topmost level is $H - 1$. The set of groups at a given level is denoted *groups(l)*.

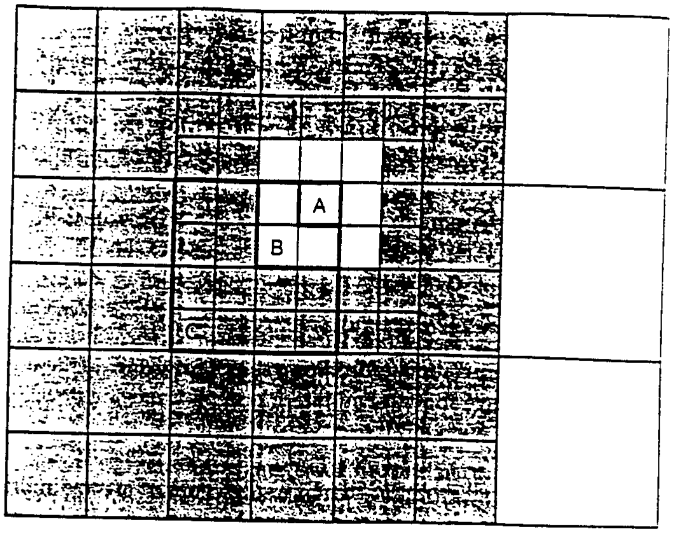


Figure 1. Multilevel FMM grouping. The small box A interacts with the dark shaded region, using the level-0 translation operators. At the next higher level, the medium box B interacts with the medium shaded region, and so on for the large box C. In general, for a low-accuracy solution ($L_l \sim k_0 D_l$), a box interacts with 27 other boxes (in 3D) through translation operators. The eight small boxes closest to A are handled directly.

The translation operators at each level will have the same number of terms L_l and far-field directions K_l because the box sizes are the same. The number of terms at each level is given by an empirical fit,[3]

$$L_l = k_0 D_l + \frac{d}{1.6} \log(k_0 D_l + \pi), \qquad (20)$$

where d is the desired number of digits and k_0 is the wave number (and should not be confused with the far-field directions). If necessary, `setup` increases the number of terms at a level until that number is a product of small primes. This makes the discrete Fourier transforms in the interpolation and filter steps fast.

For each group, `setup` constructs two lists: *nearby* and *far*. For the top group, the nearby list contains itself and the far list is empty. The routine then starts at level $l = H - 2$ and works down to $l = 0$. For each group $m \in$ *groups(l)*, it considers

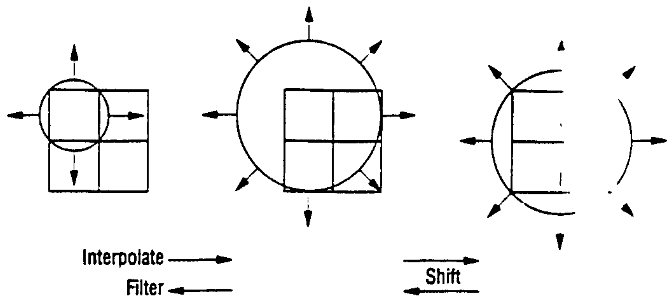


Figure 2. The interpolation-and-shifting step for moving up the tree, and the inverse shifting-and-filtering step for moving down the tree.

all groups m' that are subgroups of group in the near list of the parent group of m. If m' is far from m—that is, $k_0 X_{mm'} \geq L_l$,—`setup` places m' on *far*(m); otherwise, it places m' on *near*(m). ($X_{mm'} = X_m - X_{m'}$ where X_m is the center of group m and $X_{m'}$ is the center of group m'.) Once `setup` has constructed the near and far lists for every group in the tree, it truncates the tree (H is reduced) so that the topmost level has a reasonable number of far interactions to compute (in the full tree, groups in the topmost levels are near each other). Given the (truncated) tree with the near and far lists, `setup` can precompute all the translation operators and other needed quantities, and the routine is complete.

Apply

To form $Z \cdot I$—that is, to apply the multilevel FMM operator to the source vector—`apply` follows this procedure:

1. *Local to far*: It computes a representation of the field external to a group caused by the sources in the group. For $m \in groups(0)$,

   $$s_{mk} = \sum_{a \in sources(m)} e^{i\mathbf{k}\cdot(\mathbf{X}_m - \mathbf{x}_a)} I_a ,$$

   where a is a source in group m, $\mathbf{x}_a$ is its location, I_a is its strength, and $\mathbf{k} = k_0\hat{k}$ is a far-field direction (k_θ, k_ϕ). This shifts each source to its group's center, where its field is accumulated with that of all other sources in the group.
2. *Level-0 translation*: For $m \in groups(0)$, it computes $g_{mk} = \sum_{m' \in far(m)} T_{mm'k} s_{m'k}$, where $T_{mm'k} = \mu(k_\theta, k_\phi)$ is a level-0 translation operator as given in Equation 15, with $x_{12} = |\mathbf{X}_{m'm}|$.
3. *Uptree and translation*: Working from level $l = 1$ to $l = H - 1$, `apply` first computes the field at the center of each level l group caused by its subgroups, and then translates this field to faraway groups and accumulates the fields from subgroups. Specifically, for each subgroup m' of m, it computes $s'_{m'k} = interpolate(s_{m'k})$ and then shifts:

   $$s_{mk} = s_{mk} + e^{i\mathbf{k}\cdot(\mathbf{X}_m - \mathbf{X}_{m'})} s'_{m'k} .$$

   The interpolate step takes the external representation of the m' group ($s_{m'k}$) and converts it into a representation $s'_{m'k}$ valid for its parent group m, as we discussed in the previous section (see Equation 19). `Apply` then shifts the field $s'_{m'k}$ to the center of group m and sums that field with the contributions from the other subgroups, thereby forming an external representation of the field caused by all the sources in m. Figure 2 depicts this interpolation and shifting. The quantities s_{mk} correspond to the far-field representation of the βs in the previous section. Once `apply` has performed all the interpolation and shift steps at the level, it translates the fields, $g_{mk} = \sum_{m' \in far(m)} T_{mm'k} s_{m'k}$ for $m \in groups(l)$, using translation operators for level l. The quantities g_{mk} are the far-field representation of the αs.
4. *Downtree*: Working from level $l = H - 1$ to $l = 1$, `apply` shifts the field from each group at level l to its subgroups and converts it to the subgroup representation. Specifically, for $m \in groups(l)$ and m' a subgroup of m, it shifts

   $$g'_{mk} = e^{i\mathbf{k}\cdot(\mathbf{X}_{m'} - \mathbf{X}_m)} g_{mk}$$

   and then filters: $g_{m'k} = filter(g'_{mk})$ (see Figure 2).
5. *Far to local*: Each lowest-level group now has the field caused by all far-away groups. `Apply` computes the effect on each point in each group:

   $$B_a = \sum_k w_k g_{mk} e^{-i\mathbf{k}\cdot(\mathbf{X}_m - \mathbf{x}_a)}$$

   for $m \in groups(0)$ and $a \in sources(m)$, where w_k is the quadrature weight for the sphere rule. This corresponds to the integral over the far-field directions in Equation 16. The routine forms the quadrature weights from the product of a Gauss Legendre quadrature rule (with L_0 abscissas) in the θ (polar) angle and a trapezoidal rule (with $2L_0$ abscissas) in the ϕ (azimuthal) angle.
6. *Direct*: `Apply` directly computes the lowest-level interactions that are too close for

FMM: $B_a = B_a + \sum_{a' \in sources(m')} G(\mathbf{x}_a, \mathbf{x}_{a'})\, I_{a'}$ for $m \in groups(0)$ and $m' \in near(m)$.

The result is B with the accuracy specified by the translation operators.

Time complexity

Consider a uniform discretization of a simple convex shape, such as a sphere, having N points. The number of groups at the lowest level is $M_0 \propto N/D_0^2$. Let D_0 be one so that the groups are roughly the size of a wavelength; then $M_0 = O(N)$. For a low-accuracy solution, the number of terms at level l is $L_l \sim k_0 D_l = k_0 2^l$. The branching factor of the FMM tree for a surface is four, so the number of groups at a level is $M_l = 4^{-l} M_0$. The total number of levels, H, is then given by $M_0 = 4^{H-1}$, assuming a full tree. For $H \geq 2$, we have $H = 1 + \log_4 M_0$ and thus $H = O(\log N)$.

So, the times for the steps in `apply` are as follows:

- *Local to far*: $T_{lf} = N\, 2\, L_0^2 = 2\, k_0^2\, N = O(N)$ because the number of far-field directions at a level is $K_l = 2\, L_l^2$. The time for far to local is the same.
- *Translation*: a group interacts with 27 faraway groups (see Figure 1), which gives
$$T_t = \sum_{l=0}^{H-1} 27 M_l 2L_l^2 = 54 M_0 k_0^2 L_0^2 H = O(N \log N)\,. \tag{21}$$
- *Downtree*: To filter a single group from level l to $l-1$ requires L_l FFTs of length $2L_l$, L_{l-1} FFTs of length $2L_{l-1}$, and $2L_{l-1}$ 1D FMMs of length L_l. Recalling that each parent group must filter down to four subgroups, summing over all the levels gives
$$T_d = \sum_{l=1}^{H-1} 4M_l\big(c_a L_l 2L_l \log 2L_l + c_p 2L_{l-1} L_l \log L_l + c_a L_{l-1} 2L_{l-1} \log 2L_{l-1}\big) \tag{22}$$
$$= 8M_0 k_0^2 L_0^2 \sum_{l=1}^{H-1} \Big(c_a\big(l + \log 2k_0 L_0\big) + \big(c_p/2 + c_a/4\big)\big(l + \log k_0 L_0\big)\Big) \tag{23}$$
$$T_d = O\big(N \log^2 N\big), \tag{24}$$
where c_a and c_p are the proportionality constants for the FFT and 1D FMM. The effort in shifting is negligible. Uptree has the same order of complexity.
- *Direct*: Each lowest-level group has eight nearby groups where interactions must be handled directly (see Figure 1). So, each source has a fixed amount of work in the direct interaction that does not grow with problem size, giving a complexity of $O(N)$.

Therefore, the overall scaling for the multilevel FMM is $O(N \log^2 N)$. For higher-accuracy solutions, L_0 increases, but $L_l < 2^l L_0$ for $l > 0$, so the $O(N \log^2 N)$ scaling is an upper bound for any reasonable accuracy.

Memory

The memory required scales as $O(N \log N)$. A variety of techniques can lower the prefactor. First, because of the grouping, at a given level only a few discrete distances and orientations require translation operators. It pays to keep a cache of translation operators indexed by level, group separation ($X_{mm'}$), and the cosines that the group separator makes with two far-field directions, $(\hat{X}_{mm'} \cdot \hat{k}_1)$ and $(\hat{X}_{mm'} \cdot \hat{k}_2)$. Before `setup` computes a translation operator, it searches the cache to see if the operator has been previously computed. This results in a substantial compression of the operator, as we'll show in the next section.

Each level has only eight distinct sets of shift coefficients, which can be precomputed and stored. However, the lowest level, where individual sources are shifted to group centers and back (Steps 1 and 5), has as many coefficients as there are sources times the number of far-field directions. Precomputing these coefficients is unnecessary because they are simple exponentials. Instead, the coefficients can be computed as needed, once per `apply`. The cost of doing this can be amortized over several simultaneous operator applications. This corresponds to solving for multiple right-hand sides using a blocked iterative solver, which is a common practice. Similarly, the kernel evaluations for the direct interactions (Step 6) can be computed as needed to save memory.

Results

We implemented `apply` in C++ and ran it on an IBM RS6000/590 workstation. We used the highly optimized FFTW package for discrete Fourier transforms[13] and 1D FFM routines for filtering.[11,12] Table 1 shows the `apply` time per right-hand side and the memory requirements for spheres of increasing sizes and selected accuracies discretized by picking points randomly on the surface. Figure 3 plots the times with

Table 1. Runtime and memory requirements for `apply`, for various problem sizes and accuracies.

Points	Area λ^2	Time (seconds) for different accuracies 2	3	4	Memory use(bytes) for two-digit accuracy
314	1.26×10^2	8.30×10^{-1}	1.29×10^0	1.53×10^0	1.4×10^6
2,827	1.13×10^3	2.17×10^0	3.17×10^0	4.51×10^0	1.7×10^7
7,853	3.14×10^3	8.49×10^0	1.21×10^1	1.59×10^1	4.9×10^7
15,393	6.16×10^3	2.11×10^1	2.88×10^1	3.74×10^1	1.0×10^8
31,415	1.26×10^4	4.58×10^1	6.24×10^1	8.07×10^1	2.1×10^8
70,685	2.83×10^4	1.28×10^2	1.69×10^2	2.11×10^2	4.9×10^8
125,663	5.03×10^4	2.40×10^2	3.19×10^2	3.93×10^2	8.6×10^8
196,349	7.85×10^4	3.92×10^2	—	—	1.4×10^9

least-squares fits to the time complexity. For the two-digits case, the fit is

$$T(N) = 1.36 \times 10^{-5} N \log^2 N. \qquad (25)$$

The point at which `apply` starts to perform faster than a dense-operator application is approximately 25,000 unknowns. This assumes a sustained floating-point rate of 100 Mflops per second and no penalty for using the out-of-core techniques required to handle extremely large matrices. Table 2 shows the times for each algorithm step for the 31,415-unknowns problem.

We measured the effect of the translation operator cache, for the 196,349-unknowns problem at two-digit accuracy. On average, each level-0 translation operator is used 3,512 times; each level-1 operator is used 1,056 times; each level-2 operator is used 290 times; each level-3 operator is used 77 times; and each level-4 operator is used 14 times. The lowest levels use each operator many times because group pairs have many opportunities to be in the same relative orientation and distance. Higher levels have fewer groups and hence less potential for reuse.

Overall, the multilevel FMM memory requirements are dramatically less than that required by a dense matrix. For the 196,349-unknowns problem at two-digits accuracy, the FMM requires approximately 1.4 Gbytes, compared with the 616 Gbytes for a dense matrix (assuming double precision). This represents a savings of more than a factor of 400.

The algorithm we've described can be used to compute acoustic scattering with Dirichlet boundary conditions using a point-based, or Nyström, discretization.[14] The only additions required are that the far-to-local step must incorporate the Nyström quadrature weights and that the kernel values in the direct computation must be corrected by an appropriate scheme to accurately treat the kernel's singular nature. Many other important issues exist, such as the choice of integral-equation formulation, appropriate discretizations, and the iterative solver and preconditioner. But these are all independent of the FMM.

An extension to electromagnetic scattering or using a patch-based (Galerkin) discretization can be copied right from the single-level scheme[3] because the multilevel translation-operator machinery is independent of boundary conditions and discretizations. ♦

Acknowledgments

We thank Norman Yarvin and Vladimir Rokhlin of Yale University for supplying the efficient 1D FMM code used in the filters, John Visher of HRL for work on the filters, and Stephen Wandzura of HRL and Vladimir Rokhlin for many useful discussions. DARPA supported this work under contract MDA972-95-C-0021.

References

1. M.A. Epton and B. Dembart, "Multipole Translation Theory for the Three-Dimensional Laplace and Helmholtz Equations," *SIAM J. Scientific Computing*, Vol. 16, No. 4, July 1995, pp. 865–897.
2. V. Rokhlin, "Diagonal Form of Translation Operators for the Helmholtz Equation in Three Dimensions," *Applied and Computational Harmonic Analysis*, Vol. 1, No. 1, Dec. 1993, pp. 82–93.
3. R. Coifman, V. Rokhlin, and S. Wandzura, "The Fast Multipole Method: A Pedestrian Description,"

Table 2. Time for the algorithm steps for the 31,415-unknowns problem.

Step	Time (seconds) for given accuracy (digits)		
	2	3	4
Far ↔ local	3.6	5.9	7.2
Translation	14.5	18.1	28.5
Uptree	9.6	11.4	14.5
Downtree	11.4	13.5	16.5
Direct	6.0	12.6	12.7
Other	0.7	0.9	1.3

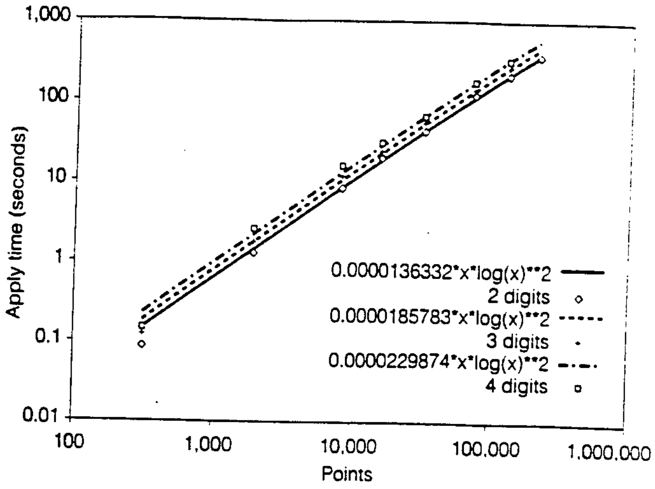


Figure 3. Time for a single FMM `apply` for accuracies of two, three, and four digits with fits to time complexity.

IEEE Antennas and Propagation Magazine, Vol. 35, No. 3, June 1993, pp. 7–12.

4. B. Dembart and E. Yip, "A 3D Moment Method Code Based on Fast Multipole," *Proc. URSI Radio Science Meeting*, Int'l Union of Radio Science, Gent, Belgium, 1994, p. 23.
5. L.R. Hamilton et al., "3D Method of Moments Scattering Computations Using the Fast Multipole Method," *1994 IEEE Antennas and Propagation Soc. Int'l Symp. Digest*, Vol. 1., IEEE Press, Piscataway, N.J., 1994, pp. 435–438.
6. L.R. Hamilton et al., "Scattering Computation Using the Fast Multipole Method," *1993 IEEE Antennas and Propagation Soc. Int'l Symp. Digest*, Vol. 2., IEEE Press, 1993, pp. 852–855.
7. B. Dembart and E. Yip, "A 3D Fast Multipole Method for Electromagnetics with Multiple Levels," *11th Ann. Rev. Progress in Applied Computational Electromagnetics*, Vol. 1, Applied Computational Electromagnetics Soc., Monterey, Calif., 1995, pp. 621–628.
8. J.M. Song and W.C. Chew, "Multilevel Fast Multipole Algorithm for Solving Combined Field Equations of Electromagnetic Scattering," *Microwave and Optical Technology Letters*, Vol. 10, No. 1, Sept. 1995, pp. 14–19.
9. M. Danos and L.C. Maximon, "Multipole Matrix Elements of the Translation Operator," *J. Math. Physics*, Vol. 6, No. 5, May 1965, pp. 766–778.
10. B.U. Felderhof and R.B. Jones, "Addition Theorems for Spherical Wave Solutions of the Vector Helmholtz Equation," *J. Math. Physics*, Vol. 28, No. 4, Apr. 1987, pp. 836–839.
11. B. Alpert and R. Jakob-Chien, *A Fast Spherical Filter with Uniform Resolution*, Tech. Report TR-50, Dept. of Computer Science and Eng., Univ. of Colorado at Denver, 1996.
12. N. Yarvin and V. Rokhlin, *A Generalized 1D Fast-Multipole Method, with Applications to Filtering of Spherical Harmonics*, tech. report, Dept. of Computer Science, Yale Univ., New Haven, Conn., to be published in 1998.
13. M. Frigo and S.G. Johnson, *The Fastest Fourier Transform in the West*, Tech. Report MIT-LCS-TR-728, Laboratory for Computer Science, Massachusetts Inst. of Technology, Cambridge, Mass., 1997; http://theory.lcs.mit.edu/~fftw.
14. L.M. Delves and J.L. Mohamed, *Computational Methods for Integral Equations*, Cambridge Univ. Press, New York, 1985.

Mark F. Gyure is a senior staff physicist in the Computational Physics Department at HRL Laboratories. His research interests are primarily in computational condensed matter and statistical physics with application to problems in materials research. He received his PhD in physics from the University of Colorado, and his MS from the University of Michigan and his BS from Ohio State University, both in aerospace engineering. He is a member of the American Physical Society and the Materials Research Society. Contact him at HRL Laboratories, M/S RL65, 3011 Malibu Canyon Rd., Malibu, CA 90265; gyure@hrl.com.

Mark A. Stalzer is a senior research scientist in the Computational Physics Department at HRL Laboratories. He received his PhD and MS in computer science from the University of Southern California and his BS in physics and computer science from California State University, Northridge. He is a member of the ACM and Sigma Chi. Contact him at HRL Laboratories, M/S RL65, 3011 Malibu Canyon Rd., Malibu, CA 90265; stalzer@hrl.com.

Accelerating Fast Multipole Methods for the Helmholtz Equation at Low Frequencies

Leslie Greengard and Jingfang Huang
Vladimir Rokhlin
Stephen Wandzura

Reprint

from

IEEE Computational Science & Engineering

Volume 5, Number 3
July-September 1998

IEEE COMPUTER SOCIETY

Washington ◆ Los Alamitos ◆ Brussels ◆ Tokyo

PUBLICATIONS OFFICE, 10662 Los Vaqueros Circle, P.O. Box 3014, Los Alamitos, CA 90720-1314 USA

© Copyright The Institute of Electrical and Electronics Engineers, Inc. Reprinted by permission of the copyright owner

Accelerating Fast Multipole Methods for the Helmholtz Equation at Low Frequencies

LESLIE GREENGARD AND JINGFANG HUANG
Courant Institute of Mathematical Sciences
VLADIMIR ROKHLIN
Yale University
STEPHEN WANDZURA
HRL Laboratories

◆ ◆ ◆

The authors describe a diagonal form for translating far-field expansions to use in low-frequency fast multipole methods. Their approach combines evanescent and propagating plane waves to reduce the computational cost of FMM implementation.

◆

Many problems in acoustics, microwave filter design, interconnect modeling, and electromagnetic scattering require the solution of the Helmholtz equation (see Figure 1). To simplify the ensuing discussion, we limit our attention to the discrete N-body problem (see Figure 1, Equation 4). The numerical difficulty here is clear; direct calculation of the sums in Equation 4 at each point requires $O(N^2)$ work, rendering large-scale calculations impractical. To overcome this obstacle, fast multipole methods have been developed over the last decade that reduce the operation count to $O(N)$ for $\omega \approx 1$ (low-frequency scattering) and $O(N \log N)$ for $\omega \approx \sqrt{N}$ (high-frequency scattering).[1–9] Still, in the 3D case, the constant implicit in the $O(N)$ notation is quite large, especially for high precision in the low-frequency regime.

We present the analytic foundations for a new version of the fast multipole method for the scalar Helmholtz equation in the low-frequency regime. The computational cost of existing FMM implementations, is dominated by the expense of translating far-field partial wave expansions to local ones, requiring $189p^4$ or $189p^3$ operations per box, where harmonics up to order p^2 have been retained. By developing a new expansion in plane waves, we can diagonalize these translation operators. The new low-frequency FMM (LF-FMM) requires $40p^2 + 6p^3$ operations per box.

For this new LF-FMM, we generalize a version of the FMM recently developed[10,11] for the Laplace equation ($\omega = 0$), which replaces the classical multipole expansion with a representation in terms of evanescent plane waves to diagonalize certain translation operators. It bears some resemblance to the FMM for the Helmholtz equation Vladimir Rokhlin developed,[1–3] which uses an expansion in terms of propagating plane waves to diagonalize translation operators. The latter method, which we will refer to as the high-frequency FMM (HF-FMM), is numerically unstable at subwavelength spatial scales. The LF-FMM we present uses a combination of evanescent and propagating modes and blends the FMM and HF-FMM together seamlessly.

1070-9924/98/$10.00 © 1998 IEEE

$$\Delta\Phi + \omega^2\Phi = f \text{ in } \Omega \subset \mathbb{R}^3 \qquad (1)$$

$$\frac{\partial\Phi}{\partial n} + \alpha\Phi = g \text{ on } \partial\Omega, \qquad (2)$$

$$r\left(\frac{\partial\Phi}{\partial r} - i\omega\Phi\right) \to 0 \text{ as } r \to \infty \qquad (3)$$

where Ω is an exterior domain and $\partial\Omega$ is its boundary. In applying integral-equation methods to Equations 1 and 2, we must repeatedly evaluate sums of the form

$$\Phi(\mathbf{x}_k) = \sum_{\substack{j=1 \\ j \neq k}}^{N} q_j \frac{e^{i\omega\|\mathbf{x}_k - \mathbf{x}_j\|}}{\|\mathbf{x}_k - \mathbf{x}_j\|} \qquad k = 1,\ldots,N, \qquad (4)$$

where the points $\mathbf{x}_k$ are in R^3, because $e^{i\omega r}/r$ is the free space Green's function for the Helmholtz equation satisfying the Sommerfeld radiation condition (Equation 3).

Figure 1. Solving the Helmholtz equation.

The multipole expansion

We now briefly define the multipole (or partial-wave) expansion due to a collection of point sources and describe some of its properties.[12-14] We will need a variety of special functions, whose definitions we collect here.

Definition 1

$P_n(x)$ denotes the Legendre polynomial of degree n, and $P_n^m(x)$ denotes the associated Legendre function of degree n and order m. Using the Rodrigues formula,

$$P_n^m(x) = (-1)^m (1 - x^2)^{m/2} \frac{d^m}{dx^m} P_n(x) \cdot$$

The spherical harmonic of degree n and order m is denoted by

$$Y_n^m(\theta,\phi) \equiv \sqrt{\frac{2n+1}{4\pi}\frac{(n-|m|)!}{(n+|m|)!}} P_n^{|m|}(\cos\theta) e^{im\phi} \,. \qquad (5)$$

We define the spherical Bessel and Hankel functions $j_n(r)$, $h_n^{(1,2)}(r)$ in terms of the usual Bessel and Hankel functions via

$$j_n(r) = \sqrt{\frac{\pi}{2r}} J_{n+1/2}(r)$$

$$h_n^{(1,2)}(r) = \sqrt{\frac{\pi}{2r}} H_{n+1/2}^{(1,2)}(r) \,.$$

Because we will always be working with Hankel functions of the first kind, we will use $h_n(r)$ as an abbreviation of $h_n^{(1)}(r)$. In particular,

$$h_0(\omega r) = \frac{e^{i\omega r}}{i\omega r} \cdot$$

Theorem 1: multipole expansion

Suppose that J sources of strengths $\{q_j, j = 1, \ldots, J\}$ are located at the points $\{\mathbf{x}_j = (\rho_j, \alpha_j, \beta_j), j = 1, \ldots, J\}$, with $|\rho_j| < a$. Then for any $\mathbf{x} = (r, \theta, \phi) \in \mathbf{R}^3$ with $r > a$, the potential

$$\Phi(\mathbf{x}) = \sum_{j=1}^{J} q_j \frac{e^{i\omega\|\mathbf{x} - \mathbf{x}_j\|}}{\|\mathbf{x} - \mathbf{x}_j\|}$$

is given by

$$\Phi(\mathbf{x}) = 4\pi\omega i \sum_{n=0}^{\infty} \sum_{m=-n}^{n} M_n^m h_n(\omega r) Y_n^m(\theta,\phi) \,, \qquad (6)$$

where

$$M_n^m = \sum_{j=1}^{J} q_j j_n(\omega\rho_j) Y_n^{-m}(\alpha_j,\beta_j) \,. \qquad (7)$$

Furthermore, for any $p \geq \omega a$,

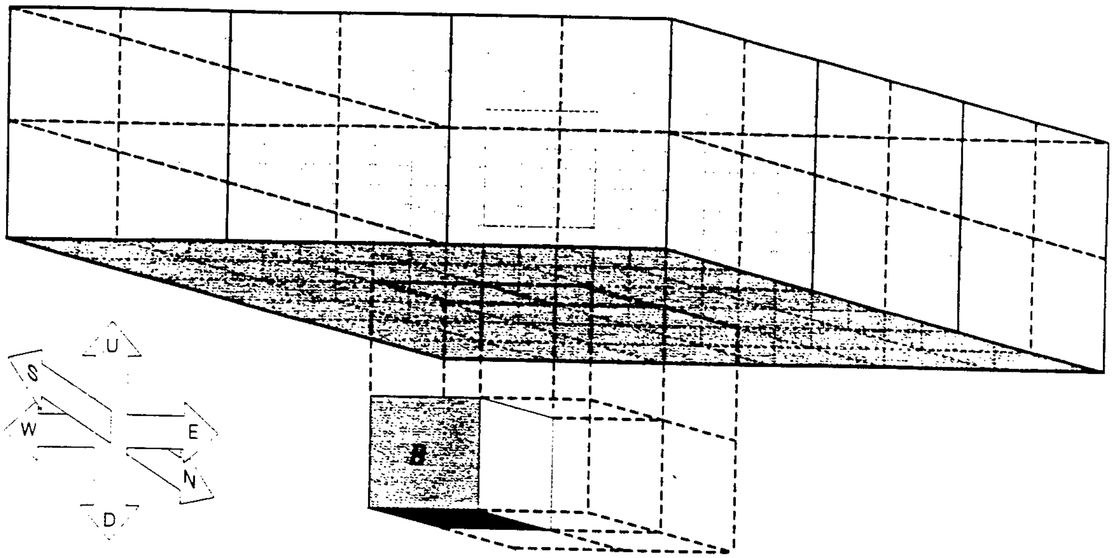
$$\left|\Phi(\mathbf{x}) - \sum_{n=0}^{p} \sum_{m=-n}^{n} M_n^m h_n(\omega r) Y_n^m(\theta,\phi)\right| = O\left(\frac{a}{r}\right)^p \,. \qquad (8)$$

Note that for Theorem 1, ωa is a measure of the radius of the enclosing sphere in terms of wavelengths. Thus, according to Equation 8, the multipole expansion does not begin to converge until the number of terms in the expansion p is of the same order as the number of wavelengths in the (smallest) enclosing sphere. Once enough terms are present, the error decay is quite rapid. Because we are interested in the low-frequency regime, we will assume that the first condition is always satisfied. If we now suppose that $r = 2a$ in the context of Theorem 1, then Equation 8 implies that

$$\left|\Phi(\mathbf{x}) - \sum_{n=0}^{p} \sum_{m=-n}^{n} M_n^m h_n(\omega r) Y_n^m(\theta,\phi)\right| = O\left(\frac{1}{2}\right)^p \,, \qquad (9)$$

and setting $p = \log_2(1/\varepsilon)$ yields a precision ε.

Figure 2. The +z list for the box *B*.



While an $N \log N$ algorithm can be constructed for the N-body problem based on only the preceding theorem, it performs poorly in 3D. The FMM relies on a more complex analysis and uses several translation operators. Because the details of such a scheme have been fully described many times,[5,6,10,15] we will not repeat them here. Instead, we will concentrate on the one translation operator whose cost is dominant in existing FMM implementations.

Theorem 2: multipole-to-local conversion

Suppose that J sources of strengths $q_1, q_2, \ldots, q_J$ are located inside a sphere of radius a with the center at the origin. Suppose also that $Q = (\rho, \alpha, \beta)$, and that $\rho > (c + 1)a$ with $c > 1$. Then the multipole expansion (Equation 6) converges inside the sphere D_Q of radius a centered at Q. Inside D_Q, the potential due to the charges $q_1, q_2, \ldots, q_J$ is described by a local expansion:

$$\Phi(\mathbf{x}) = \sum_{l=0}^{\infty} \sum_{k=-l}^{l} L_l^k j_l(\omega r') Y_l^k(\theta', \phi') \quad , \tag{10}$$

where (r', θ', ϕ') are the coordinates of $\mathbf{x}$ with respect to the center Q. Furthermore, for any $p \geq \omega a$,

$$\left| \Phi(\mathbf{x}) - \sum_{l=0}^{p} \sum_{k=-l}^{l} L_l^k j_l(\omega r) Y_l^k(\theta, \phi) \right| = O\left(\frac{1}{c}\right)^p \quad . \tag{11}$$

For Theorem 2, the matrix that converts the multipole coefficients $\{M_n^m\}$ into the local coefficients $\{L_l^k\}$ is rather complicated,[5,16] and we omit it. We simply observe here that the matrix is dense, so applying it to a truncated expansion with $O(p^2)$ harmonics requires $O(p^4)$ work.

Although, as indicated above, we will not describe the full 3D fast multipole algorithm, it is based on a hierarchical subdivision of space. For this, we assume that all sources are contained in a box of side length D, which we refer to as refinement level 0. We obtain refinement level $l + 1$ recursively from level l by subdividing each box into eight equal parts. This yields a natural tree structure, where the eight boxes at level $l + 1$ obtained by subdividing a box at level l are considered its children. Below we define boxes at the same refinement level (Definitions 2 and 3) as well as the interaction list associated with each box (Definition 4).

Table 1. The interaction list for a box *B* is subdivided into six lists, one associated with each direction.

Interaction list	Elements
+*z* list	Separated by at least one box in the +*z* direction
−*z* list	Separated by at least one box in the −*z* direction
+*y* list	Separated by at least one box in the +*y* direction and not contained in the +*z* or −*z* lists
−*y* list	Separated by at least one box in the −*y* direction and not contained in the +*z* or −*z* lists
+*x* list	Separated by at least one box in the +*x* direction and not contained in the +*z*, −*z*, +*y*, or −*y* lists
−*x* list	Separated by at least one box in the −*x* direction and not contained in the +*z*, −*z*, +*y*, or −*y* lists

- *Definition 2.* Two boxes are said to be *near neighbors* if they are at the same refinement level and share a boundary point (a box is a near neighbor of itself).
- *Definitions 3.* Two boxes are said to be *well separated* if they are at the same refinement level and are not near neighbors.
- *Definition 4.* With each box i is associated an *interaction list*, consisting of the children of the near neighbors of i's parent which are well separated from box i.

A simple counting argument shows that the interaction list contains up to 189 boxes. In the FMM, the most expensive step is converting the multipole expansion for each box into the 189 different local expansions that the boxes in its interaction list require. If there are M boxes in the hierarchy, then this requires $O(189p^4M)$ work.

Diagonal form of translation operators

The new generation of FMMs is based on combining multipole expansions with exponential or plane-wave expansions. A complicating feature of this approach, however, is that we need six different expansions for each box, one emanating from each face of the cube. The interaction list for each box is subdivided into six lists, one associated with each direction. Figure 2 shows the $+z$ list for the box B, and Table 1 explains the six lists for the interaction list. After reviewing Table 1, it is easy to verify that the original interaction list is equal to the union of the $+z$, $-z$, $+y$, $-y$, $+x$, and $-x$ lists.

The starting point for our analysis is the integral representationrepresentation

$$\frac{e^{i\omega r}}{r} = \frac{1}{2\pi}\int_0^\infty e^{-\sqrt{\lambda^2-\omega^2}\,z}\int_0^{2\pi} e^{i\lambda(x\cos\alpha+y\sin\alpha)}\frac{\lambda}{\sqrt{\lambda^2-\omega^2}}\,d\alpha\, d\lambda\quad, \qquad (12)$$

which is valid for $z > 0$. It is straightforward to derive from the 3D Fourier transform of the kernel $e^{i\omega r}/r$, followed by contour integration. We need the restriction $z > 0$ for the contour integral to be well-defined.[14] The 2D formula is given in the "2D Fourier transform" sidebar.

Note that, for $0 \le \lambda \le \omega$, the modes propagate without attenuation, while for $\omega \le \lambda \le \infty$, they decay. We refer to the first region as the *propagating* part of the spectrum and the second as the *evanescent* part. For the propagating part, we change variables $\lambda = \omega \sin\theta$ (see Figure 3 for the resulting equation). For the evanescent part, we change variables $\sigma^2 = \lambda^2 - \omega^2$ (see Figure 4 for the resulting equation).

$$\left(\frac{e^{i\omega r}}{r}\right)_{prop} = \frac{1}{2\pi}\int_0^\omega e^{-\sqrt{\lambda^2-\omega^2}\,z}\int_0^{2\pi} e^{i\lambda(x\cos\alpha+y\sin\alpha)}\frac{\lambda}{\sqrt{\lambda^2-\omega^2}}\,d\alpha\, d\lambda$$
$$= \frac{\omega}{2\pi i}\int_0^{\pi/2} e^{i\omega\cos\theta\, z}\int_0^{2\pi} e^{i\omega\sin\theta(x\cos\alpha+y\sin\alpha)}\,d\alpha\,\sin\theta\, d\theta \qquad (13)$$

Figure 3. For the propagating part of the spectrum, we change variables $\lambda=\omega\sin\theta$ (Equation 13).

$$\left(\frac{e^{i\omega r}}{r}\right)_{evanescent} = \frac{1}{2\pi}\int_\omega^\infty e^{-\sqrt{\lambda^2-\omega^2}\,z}\int_0^{2\pi} e^{i\lambda(x\cos\alpha+y\sin\alpha)}\frac{\lambda}{\sqrt{\lambda^2-\omega^2}}\,d\alpha\, d\lambda$$
$$= \frac{1}{2\pi}\int_0^\infty e^{-\sigma z}\int_0^{2\pi} e^{i\sqrt{\sigma^2+\omega^2}(x\cos\alpha+y\sin\alpha)}\,d\alpha\, d\sigma$$
$$= \frac{1}{2\pi}\int_0^\infty e^{-\sigma z} J_0\left(\sqrt{\sigma^2+\omega^2}\sqrt{x^2+y^2}\right)d\sigma \qquad (14)$$

Figure 4. For the evanescent part of the spectrum, we change variables $\sigma^2 = \lambda^2 - \omega^2$ (Equation 14).

In Equation 12, as $\omega \to 0$, the propagating part disappears, leaving only the evanescent spectrum. This is the integral representation for $1/r$ used in

2D Fourier transform

In 2D, the analog of Equation 12 (see the main text) is

$$H_0(\omega r) = \frac{1}{\pi}\int_{-\infty}^{\infty}\frac{e^{i\lambda x}e^{-\sqrt{\lambda^2-\omega^2}\,y}}{\sqrt{\omega^2-\lambda^2}}\,d\lambda\quad, \qquad (23)$$

which is valid for $y>0$.

The propagating part, as above, covers the range $|\lambda| \le \omega$. Using the change of variables $\lambda = \omega\cos\theta$ yields

$$(H_0(\omega r))_{prop} = \frac{1}{\pi}\int_0^\pi e^{-i\omega(x\cos\theta-y\sin\theta)}\,d\theta\quad. \qquad (24)$$

For the evanescent part, we make the change of variables $\sigma^2 = \lambda^2 - \omega^2$, so that

$$(H_0(\omega r))_{evanescent} = \frac{1}{\pi i}\int_{-\infty}^{\infty}\frac{e^{-\sigma y}e^{i\sqrt{\sigma^2+\omega^2}\,x}}{\sqrt{\sigma^2-\omega^2}}\,d\sigma\quad. \qquad (25)$$

the FMM—hence our assertion that we have a seamless transition to the zero frequency case.

The next problem we face is that of discretization. The integrand for the propagating part is smooth, and we achieve high-order accuracy via Gaussian quadrature in the θ direction and the trapezoidal rule in the α direction. The evanescent part is more complicated. The inner integral, with respect to α, is easily handled by the trapezoidal rule (which achieves spectral accuracy for periodic functions), but the outer integral requires more care. We use generalized Gaussian quadrature rules,[17] designed with the geometry of the interaction list in mind. We present our analysis in the "Discretization" sidebar.

Incorporation into LF-FMM

Consider now the interaction list for a box B in the context of a fast multipole code, for which we need $189p^4$ operations with the naive multipole-to-local translation operator, and $189p^3$ operations using rotation matrices.[10] Using the analysis outlined in the "Discretization" sidebar, we can generate six outgoing exponential expansions at a cost of $6p^3$ work and translate them all at a cost of $189p^2$ work. Once a box has received the incoming exponential expansions from all directions, it can convert them to a single local expansion, using an additional $6p^3$ operations. Thus, the total work scales like $12p^3 + 189p^2$ operations per box. Further symmetry considerations reduce this to $6p^3 + 40p^2$ operations.[10]

Discretization

Because of the restriction that $z > 0$, we assume, for the moment, that a source $Q = (x_0, y_0, z_0)$ is contained in a box B and that a target $P = (x, y, z)$ lies in a box $C \in +z - list(B)$. To fix spatial scales, we assume that B and C have unit volume and that they are separated in the z-direction by one or two unit distances. We then have the following result.[1]

Lemma 1: plane wave representation

Let r_{PQ} denote the distance from $Q \in B$ to $P \in C \in +z - list(B)$, and let $\{\theta_1, \ldots, \theta_N\}$ and $\{v_1, \ldots, v_N\}$ be the nodes and weights for N-point Gauss-Legendre quadrature on the interval $[0, \pi/2]$. Then there exist weights $\mu_1, \ldots, \mu_s$, nodes $\sigma_1, \ldots, \sigma_s$, and integers $M(1), \ldots, M(s)$, so that

$$\left| \frac{e^{i\omega r_{PQ}}}{r_{PQ}} - \omega \sum_{k=1}^{N} \frac{\upsilon_k \sin\theta_k}{iN} \sum_{j=1}^{N} e^{i\omega\left[-\cos\theta_k (z-z_0) + \sin\theta_k \left[(x-x_0)\cos\alpha_j + (y-y_0)\sin\alpha_j\right]\right]} \right.$$
$$\left. - \sum_{k=1}^{s} \frac{\mu k}{M(k)} \sum_{j=1}^{M(k)} e^{-\sigma_k (z-z_0) + i\sqrt{\sigma_k^2 + \omega^2}\left[(x-x_0)\cos\alpha_j + (y-y_0)\sin\alpha_j\right]} \right| < \varepsilon \qquad (15)$$

for $0 \le \omega\, r_{PQ} \le 10$, where $\alpha_j = 2\pi j/M(k)$. The total number of exponentials required, which we denote by S_{exp}, satisfies

$$S_{exp} = N^2 + \sum_{k=1}^{s} M(k) = O(\log_2^2 \varepsilon)\,.$$

Norman Yarvin and Vladimir Rokhlin supply us with the weights and nodes μ_i and σ_i for the evanescent modes.[2] For six-digit accuracy, the total number of modes we require is approximately 600—150 for the propagating spectrum and 450 for the evanescent spectrum. Ten-digit accuracy requires 1,500 modes—300 for the propagating spectrum and 1,200 for the evanescent spectrum. (The FMM for the Laplace equation requires 280 modes at six-digit accuracy and 900 modes at 10-digit accuracy.)

Corollary 1

Let B be a box of unit volume centered at the origin containing L sources of strengths $\{q_l, l = 1,\ldots,L\}$, located at the points $\{Q\} = (x_l, y_l, z_l), l = 1,\ldots,L\}$. Then for any P contained in $+z - list(B)$, the potential $\Phi(P)$ satisfies

$$\left| \Phi(P) - \omega \sum_{k=1}^{N} \sum_{j=1}^{N} W^P(k,j) e^{-i\omega\cos\theta_k z} e^{i\omega\sin\theta_k \cos\alpha_j x} e^{i\omega\sin\theta_k \sin\alpha_j y} \right.$$
$$\left. - \sum_{k=1}^{s} \sum_{j=1}^{M(k)} W^E(k,j) e^{-\sigma_k z} e^{i\sqrt{\sigma_k^2+\omega^2}\cos\alpha_j x} e^{i\sqrt{\sigma_k^2+\omega^2}\sin\alpha_j y} \right| < A\varepsilon \qquad (16)$$

Significant implementation work remains, including the coupling of this LF-FMM with an HF-FMM, once the dimensions of a box are on the order of a wavelength. Current HF-FMM implementations have been able to investigate structures that are many wavelengths across, but only those with smooth surfaces. A hybrid code will be able to include subwavelength mesh refinement and will greatly enhance the range of future simulation efforts. ♦

where $A = \sum^{L}_{l=1} |q_l|$,

$$W^P(k,j) = \frac{\upsilon_k \sin\theta_k}{iN}\sum_{l=1}^{L} q_l e^{i\omega\cos\theta_k z_l} e^{-i\omega\sin\theta_k\cos\alpha_j x_l} e^{-i\omega\sin\theta_k\sin\alpha_j y_l} \tag{17}$$

and

$$W^E(k,j) = \frac{\mu_k}{M(k)}\sum_{l=1}^{L} e^{\sigma_k z_l} e^{-i\sqrt{\sigma_k^2+\omega^2}\cos\alpha_j x_l} e^{-i\sqrt{\sigma_k^2+\omega^2}\sin\alpha_j y_l} \quad . \tag{18}$$

Corollary 2: Diagonal translation

Let B be a box of unit volume centered at the origin containing N charges of strengths $\{q_l, l = 1, \ldots, L\}$, located at the points $\{Q_l = (x_l, y_l, z_l), = 1, \ldots, L\}$ and let C be a box in $+z$–*list*(B) centered at (x_c, y_c, z_c). For $P \in C$, let the potential $\Phi(P)$ be approximated by the exponential expansion centered at the origin

$$\Phi(P) = \omega\sum_{k=1}^{N}\sum_{j=1}^{N} W^P(k,j)e^{-i\omega\cos\theta_k z} e^{i\omega\sin\theta_k(\cos\alpha_j x+\sin\alpha_j y)}$$
$$+\sum_{k=1}^{s}\sum_{j=1}^{M(k)} W^E(k,j)e^{-\sigma_k z} e^{i\sqrt{\sigma_k^2+\omega^2}(\cos\alpha_j x+\sin\alpha_j y)} + O(\varepsilon) \tag{19}$$

Then

$$\Phi(P) = \omega\sum_{k=1}^{N}\sum_{j=1}^{N} V^P(k,j)e^{-i\omega\cos\theta_k(z-z_c)} e^{i\omega\sin\theta_k(\cos\alpha_j(x-x_c)+\sin\alpha_j(y-y_c))}$$
$$+\sum_{k=1}^{s}\sum_{j=1}^{M(k)} V^E(k,j)e^{-\sigma_k(z-z_c)} e^{i\sqrt{\sigma_k^2+\omega^2}(\cos\alpha_j(x-x_c)+\sin\alpha_j(y-y_c))}$$
$$+O(\varepsilon) \tag{20}$$

where

$$V^P(k,j) = W^P(k,j)e^{-i\omega\cos\theta_k z_c} e^{i\omega\sin\theta_k\cos\alpha_j x_c} e^{i\omega\sin\theta_k\cos\alpha_j y_c} \tag{21}$$

and

$$V^E(k,j) = W^E(k,j)e^{-\sigma_k z_c} e^{i\sqrt{\sigma_k^2+\omega^2}\cos\alpha_j x_c} e^{i\sqrt{\sigma_k^2+\omega^2}\sin\alpha_j y_c} \quad . \tag{22}$$

Equations 21 and 22 are, in some sense, the centerpiece of the new scheme. They show that p^2 degrees of freedom describing the far field due to sources in a box B can be transmitted to a box C in its interaction list using p^2 operations. In other words, in a plane-wave expansion, translation is equivalent to multiplication (see Figure A).

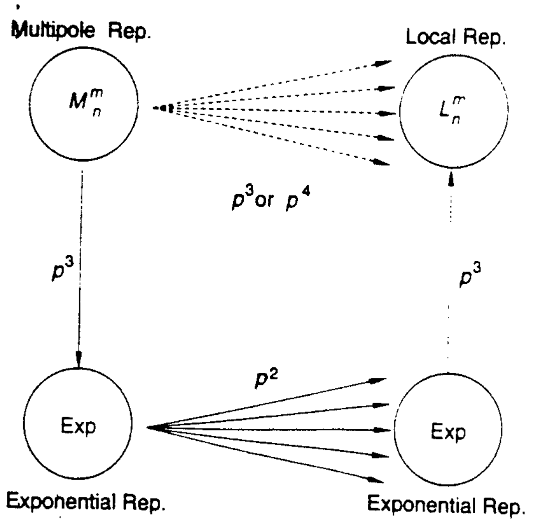


Figure A. In the new FMM, we can replace a large number of multipole-to-local translations—costing $O(p^3)$ or $O(p^4)$ work—with a large number of exponential translations, costing $O(p^2)$ work.

In an actual FMM implementation, we will be given the multipole expansion for a box B rather than the source distribution itself, so we will need to convert it to an exponential expansion. Moreover, after translating an exponential expansion, we must convert it to a local harmonic expansion of the form (see Equation 10 in the main text). The formulae are rather complex, and we avoid going into detail.[2] Here, we simply observe that $O(p^3) = O(\log^3 \varepsilon)$ work is required for each step.

Up to this point, we have considered only the exponential expansion needed for the $+z$ list. To obtain expansions appropriate for each of the other five lists, we simply rotate the coordinate system so that the z axis points in the desired direction. The cost of rotation also scales as $O(p^3)$.

References

1. J. Huang and L. Greengard, "Diagonal Forms of Translation Operators for Helmholtz and Yukawa Potentials," *in preparation.*
2. N. Yarvin and V. Rokhlin, *Generalized Gaussian Quadratures and Singular Value Decompositions of Integral Operators*, Tech. Report 1109, Computer Science Dept., Yale Univ., New Haven, Conn., 1996.

Acknowledgements

The work of Leslie Greengard and Jingfang Huang was supported in part by the Applied Mathematical Sciences Program of the U.S. Department of Energy under Contract DE-FGO288ER25053. The work of Leslie Greengard and Vladimir Rohklin was supported in part by DARPA/AFOSR under Contract F49620-95-C-0075. Rohklin was also supported in part by ONR under Grant N00014-96-1-0188. Steve Wandzura's work was supported by DARPA/DSO under contract MDA972-95-C-002.

References

1. R. Coifman, V. Rokhlin, and S. Wandzura, "The Fast Multipole Method for the Wave Equation: A Pedestrian Prescription," *IEEE Antennas and Propagation*, Vol. 35, No. 7, 1993.
2. V. Rokhlin, "Rapid Solution of Integral Equations of Scattering Theory in Two Dimensions," *J. Computational Physics*, Vol. 86, 1990, pp. 414–439.
3. V. Rokhlin, "Diagonal Forms of Translation Operators for the Helmholtz Equation in Three Dimensions," *Applied and Computational Harmonic Analysis*, Vol. 1, Academic Press, San Diego, 1993, pp. 82–93.
4. F.X. Canning, "Sparse Approximation for Solving Integral Equations with Oscillatory Kernels," *SIAM J. Scientific Statistical Computing*, Vol. 13, 1992, pp. 71–87.
5. M.A. Epton and B. Dembart, "Multipole Translation Theory for Three-Dimensional Laplace and Helmholtz Equations," *SIAM J. Scientific Computing*, Vol. 16, 1995, pp. 865–897.
6. L. Greengard, *The Rapid Evaluation of Potential Fields in Particle Systems*, MIT Press, Cambridge, Mass., 1988.
7. L. Greengard, "Fast Algorithms for Classical Physics," *Science*, Vol. 265, 1994, p. 909.
8. L. Greengard and V. Rokhlin, "Rapid Evaluation of Potential Fields in Three Dimensions," *Vortex Methods*, C. Anderson and C. Greengard, eds., *Lecture Notes in Mathematics*, No. 1360, Springer-Verlag, New York, 1988, pp. 121.
9. J.M. Song and W.C. Chew, "Multilevel Fast Multipole Algorithm for Solving Combined Field Integral Equations of Electromagnetic Scattering," *Microwave and Optical Technolnology Letters*, Vol. 10, No. 1, 1995, pp. 14–19.
10. L. Greengard and V. Rokhlin, "A New Version of the Fast Multipole Method for the Laplace Equation in Three Dimensions," *Acta Numerica*, Vol. 6, 1997, pp. 229–269.
11. T. Hrycak and V. Rokhlin, *An Improved Fast Multipole Algorithm for Potential Fields*, Report 1089, Dept. of Computer Science, Yale Univ., 1995.
12. M. Abramowitz and I. Stegun, eds., *Handbook of Mathematical Functions*, Dover, New York, 1965.
13. J. D. Jackson, *Classical Electrodynamics*, Wiley, New York, 1975.
14. P.M. Morse and H. Feshbach, *Methods of Theoretical Physics*, McGraw-Hill, New York, 1953.
15. K. Nabors et al., "Preconditioned, Adaptive, Multipole-Accelerated Iterative Methods for Three-Dimensional First-Kind Integral Equations of Potential Theory," *SIAM J. Scientific Statistical Computing*, Vol. 15, 1994, p. 714.
16. M. Danos and L.C. Maximon, "Multipole Matrix Elements of the Translation Operator," *J. Math. Phys.* Vol. 6, 1965, pp. 766–778.
17. N. Yarvin and V. Rokhlin, *Generalized Gaussian Quadratures and Singular Value Decompositions of Integral Operators*, Tech. Report 1109, Computer Science Dept., Yale Univ., 1996.

Leslie Greengard's bio appears on p. 18.

Jingfang Huang is a postdoctural fellow at the Courant Institute of Mathematical Sciences, New York University. His research interests include scientific computing, numerical methods, and fast algorithms and their applications to partial differential equations. He received his BSc in applied mathematics from Tsinghua University, Beijing, and his MA and PhD from the Courant Institute of Mathematical Sciences. Contact him at the Courant Inst. of Mathematical Sciences, NYU, 251 Mercer St., New York, NY 10012.

Vladimir Rokhlin is a professor of computer science and mathematics at Yale University. His research is in the areas of numerical scattering theory, elliptic partial differential equations, numerical solution of integral equations, quadrature formulas for singular functions, and numerical complex analysis. He received his MS in mathematics from Vilnius University, Lithuania, and his PhD in applied mathematics from Rice University. Contact him at the Department of Computer Science, Yale Univ., New Haven, CT 06250.

Stephen Wandzura's bio appears on p. 18.

There are over

90,000 REASONS

to join the IEEE COMPUTER SOCIETY

With a huge and worldwide membership, the IEEE Computer Society offers you unparalleled access to the best minds in computer science.

THE IEEE COMPUTER SOCIETY is not just the leading provider of technical information and services to the world's computing professionals. It's also the oldest and largest association of computer professionals in the world, offering over 90,000 members a comprehensive program of publications, meetings, and technical activities. It's that networking that provides one of the biggest benefits of membership. There are other benefits, too.

http://computer.org

The society Web site contains a great deal of information and provides additional services to members. Society members automatically receive electronic access to all issues of *Computer* magazine from 1995 forward along with their paper issues. Members may establish a free email alias @computer.org. Only members may choose to subscribe to electronic versions of our optional periodicals – getting full-text, searchable access to all issues from 1995 forward. Members may choose to network through the CS Member Network, where you can post your technical interests and search on other members interest areas, employers, or geographies.

COMPUTER SOCIETY PRESS

As a member you'll have access to the best periodicals, books, and other materials at low prices. There are over 300 titles to choose from, covering a broad spectrum of topics. The Computer Society press publishes books, periodicals, tutorials, conference proceedings, executive briefings, and CD-ROMs.

CONFERENCES, SYMPOSIA, WORKSHOPS

The society sponsors or cosponsors over 110 prestigious technical workshops, symposia and conferences. You'll be able to get together with peers, colleagues, and ultimately, friends at the premier technical meetings in the computer science and engineering profession.

TECHNICAL COMMITTEES AND CHAPTER ACTIVITIES

TC's are the heart of the Computer Society. You're invited to join and participate in up to four of the society's 32 technical committees or task forces at no extra cost. And every member may take advantage of their local chapter activities with more than 300 regular and student chapters worldwide.

STANDARDS ACTIVITIES

The IEEE Computer Society is a leader in developing standards with significant impact on the computing industry and therefore, the world. There are 12 standards committees, with over 200 work groups attached to them. All society members are invited to participate.

AWARDS AND PROFESSIONAL RECOGNITION

The society sponsors an active and prestigious awards program and participates in nominations for IEEE awards and the highly respected designation of IEEE Fellow.

NOW IS THE TIME TO JOIN

You might not have considered joining a society before. But it's easy to join, it's totally inexpensive, the industry is rapidly changing and expanding, and as you can see, there are many benefits. Visit our home page at http://computer.org or contact us at the office nearest you to join today.

EUROPE
IEEE Computer Society
13, Avenue de l'Aquilon
B-1200 Brussels, Belgium
Ph: +32-2-770-2198
Fx: +32-2-770-8505
Email: euro.ofc@computer.org

ASIA/PACIFIC RIM
IEEE Computer Society
Ooshima Bldg
2-19-1 Minami Aoyama
Minato-ku, Tokyo 107, Japan
Ph: +81-3-3408-3118
Fx: +81-3-3408-3553
Email: tokyo.ofc@computer.org

ALL OTHERS
IEEE Computer Society
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314, USA
Ph: 1-714-821-8380
Fx: 1-714-821-4641
Email: membership@computer.org

http://computer.org

THE INSTITUTE OF ELECTRICAL
& ELECTRONICS ENGINEERS, INC.
IEEE

IEEE
COMPUTER
SOCIETY

the WORLD'S
COMPUTER
SOCIETY

Information about the IEEE Computer Society and its services is available by calling our Customer Service Department at **+1-714-821-8380** or by e-mail: **cs.books@computer.org**. The Society maintains its home page on the World Wide Web at **http://computer.org**

IEEE COMPUTER SOCIETY

A Scalable Multilevel Helmholtz FMM for the Origin 2000*

Mark A. Stalzer†

Abstract

Presented is a parallel algorithm based on the multilevel fast multipole method (FMM) for the Helmholtz equation. This variant of the FMM is useful for electromagnetic scattering calculations. The algorithm was implemented on an SGI Origin 2000 using a threaded approach without explicit message passing. To achieve good scalability, steps in the FMM that intrinsically require inter-processor communications (applying far field translation operators) were modified to improve cache performance and minimize communications costs.

1 Introduction

This paper presents a scalable parallel version of the multilevel fast multipole method (FMM) for the Helmholtz equation: $(\nabla^2 + k^2)\Psi = \rho$. This variant of the FMM is useful for computing scattering cross sections and antenna radiation patterns[2, 3, 5, 6]. This is in contrast to the FMM for the Laplace equation, $\nabla^2\Psi = \rho$, which is applicable to the N-body problem. A substantial amount of work has been done on parallelizing the (multilevel) Laplace FMM[4, 8, 10], and single-level Helmholtz FMM[7, 9]. The emphasis here is on a scalable parallel *multilevel* Helmholtz FMM.

This paper is organized as follows. In the next section, the basics of the multilevel Helmholtz FMM are reviewed. In Section 3, the computation model is presented followed by the details of the parallel FMM implementation in Section 4. Scalability results are given in Section 5 followed by some concluding remarks.

2 Fast Multipole Method

A method of frequent choice for computing scattering cross sections and radiation patterns is to solve a matrix equation, $Z \cdot I = V$, derived from the discretization of an integral equation. The number of unknowns N required for accurate modeling of such problems can be very large, which severely limits problem size. The system can be solved by factoring the dense matrix Z (an $O(N^3)$ operation), or by using an iterative technique which requires $O(N^2)$ operations per iteration. The $O(N^2)$ operation in iterative solvers is the multiplication of an approximation $\tilde{I}$ by the impedance matrix Z. In contrast, the FMM works by recursively decomposing Z into sparse components that can be applied in $O(N \log^2 N)$ time.

The basic approach given here follows the paper by Gyure and Stalzer[5]. Consider two well-separated spheres of radius R_1 and R_2, each containing a collection of Helmholtz sources. The field due to an individual source is given by

$$\phi(\mathbf{r}) = G(\mathbf{r}) = \frac{e^{ik_0 r}}{k_0 r} \tag{1}$$

---

*This work was supported by DARPA under contract MDA972-95-C-0021 and the Hughes Electronics Corporation.

†HRL Laboratories, Malibu, CA

where $\mathbf{r}$ is relative to the source and k_0 is the free space wavenumber. (Given a vector $\mathbf{r}$, r is its magnitude and $\hat{r}$ is the corresponding unit vector.) We want to quickly evaluate the field generated by all the sources in R_1 at every source in R_2. This field can be written as a multipole expansion valid outside of R_1 as

$$\psi(\mathbf{r}) = \sum_{lm} \beta_{lm} h_l(kr) Y_{lm}(\theta, \phi) \tag{2}$$

where r, θ, and ϕ are relative to a coordinate system centered in R_1. $h_l(kr)$ are spherical Hankel functions of the first kind, and $Y_{lm}(\theta, \phi)$ are normalized spherical harmonics. We'll refer to this expansion as an h-expansion. Similarly, we can write an expression for the field valid inside R_2 :

$$\varphi(\mathbf{r}) = \sum_{lm} \alpha_{lm} j_l(kr) Y_{lm}(\theta, \phi) \tag{3}$$

where the coordinate system is now centered in R_2, and $j_l(kr)$ are spherical Bessel functions. We'll refer to this expansion as a j-expansion. For the moment, we consider both of these to be infinite sums. The FMM then rests on three observations:

- The origin of an h-expansion can be shifted arbitrarily inside R_1, and a new set of coefficients, $\tilde{\beta}_{lm}$, can be computed for this new expansion. The same holds for shifting a j-expansion arbitrarily to a new origin inside of R_2, which results in a new set of coefficients, $\tilde{\alpha}_{lm}$.
- An h-expansion valid outside of R_1 can be translated and converted into a j-expansion valid inside R_2, resulting in a new set of coefficients for the j-expansion, γ_{lm}.
- Most crucial, these shifts and translations can be done efficiently by transforming the coefficients into a basis in which both operators are diagonal.

The *far field* transform of an arbitrary function $f(\hat{k})$ is

$$f(\hat{k}) = \sum_{lm} i^l Y_{lm}(\hat{k}) f_{lm} \tag{4}$$

and the inverse transform is given by

$$f_{lm} = \int d\hat{k} i^{-l} Y^{\star}_{lm}(\hat{k}) f(\hat{k}) \tag{5}$$

where $\hat{k}$ is a unit vector represented by polar and azimuthal angular components: (k_θ, k_ϕ).

It is in this k-basis that the shift and translation operators are diagonal. An h-expansion in its far-field basis is shifted from a point $\mathbf{x}$ to another point $\mathbf{x}'$ both inside of R_1 by

$$\tilde{\beta}(\hat{k}) = \lambda(\hat{k}, \mathbf{x}' - \mathbf{x}) \beta(\hat{k}) \tag{6}$$

where λ is given by

$$\lambda(\hat{k}, \mathbf{x}' - \mathbf{x}) = e^{ik_0 \hat{k} \cdot (\mathbf{x}' - \mathbf{x})} \tag{7}$$

The same shift operator λ also applies to j-expansions. It represents a "local" shift in the group center, retaining the exterior or interior expansion.

The translation of an h-expansion into a j-expansion is through the translation operator μ, which, in the far-field basis is

$$\mu(\hat{k}, \mathbf{x}' - \mathbf{x}) = \sum_{l} i^l (2l+1) h_l(k_0 |\mathbf{x}' - \mathbf{x}|) P_l(\hat{k} \cdot (\mathbf{x}' - \mathbf{x}) / |\mathbf{x}' - \mathbf{x}|) \tag{8}$$

where the P_l are Legendre polynomials.

In practice the expansions are truncated to a finite number of terms L depending on the group size and desired accuracy. The mathematical validity of this truncation is addressed by Rokhlin[6] but it is related to the fact that these series are asymptotic and are, therefore, of controllable accuracy. Empirically, it has been determined that the number of terms L needed in the expansions for a region of diameter D is[2]

$$L = k_0 D + \frac{d}{1.6}\log(k_0 D + \pi) \tag{9}$$

where d is the desired number of digits.

The above expressions for the translation operators, together with the far field transform, are the basic tools used to construct a multilevel FMM algorithm. Clearly the field caused by a collection of sources inside an arbitrary group G_1 can be evaluated at any point inside a second group G_2 by converting the exterior h-expansion, valid outside G_1, to an interior j-expansion which is valid inside G_2. Also, we can calculate the field at that point caused by the sources in G_1 by computing $\tilde{\alpha}_{00}$, the leading term in the j-expansion. No other terms contribute, because the expansion is already centered at the field point where $r = 0$ and all the terms $j_l(0)$ are zero except for j_0 which is one. Thus, we can evaluate the field directly through the far-field transform as

$$\phi(0) = \tilde{\alpha}_{00} = \frac{1}{\sqrt{4\pi}} \int d\hat{k}\tilde{\alpha}(\hat{k}). \tag{10}$$

The abcissae $\hat{k} = (k_\theta, k_\phi)$ of the numerical quadrature rule used to compute this integral are selected so that it can be performed exactly. One choice is to use a trapezoidal rule of $2L$ points in the ϕ direction and an L point Gauss-Legendre rule in the θ direction. This discretization of the $\hat{k}$ basis is used throughout the FMM.

The multilevel Helmholtz FMM works in fundamentally the same way as the Laplace FMM in that it combines expansions valid inside the original groups to form expansions valid inside correspondingly larger groups with bigger group diameters. This recursive regrouping results in a tree-like structure that has groups of different sizes at different levels of the tree. The h-expansions from neighboring groups are shifted and combined into a single h-expansion representing a larger group when going up the tree, and j-expansions in a large group are converted to smaller groups going down the tree. The details of this process are given in the next section.

There is, however, an important mathematical detail. When going up the tree, it is necessary to interpolate the far field representation of a group at one level onto the denser ($\hat{k}$ more closely spaced) basis of the group one level higher. Similarly, when going down the tree, it is necessary to convert to a sparser basis in a filtering process. In both cases, the code converts from the far field basis to the multipole coefficients and then back to the new far field basis using the definitions given in Equations 4 and 5. The actual implementation is in terms of fast Fourier transforms for the k_ϕ direction, and fast associated Legendre transforms for the k_θ direction[11]. As a practical matter, a slow associated Legendre transform which is implemented in terms of matrix multiplication can be used on rather large problems because of the small prefactor in its time complexity relative to the fast transform. However, fetching the transform matrices from memory causes some scalability problems which are addressed in Section 4.2. The details of the filtering and interpolation processes are given in [5].

3 Computation Model

The parallel FMM is implemented using threads assuming a cache-coherent distributed shared memory mechanism such as that on the Origin 2000. The O2000 is constructed as a collection of nodes interconnected by a hypercube. A node consists of two processors, each with two levels of cache, and a local memory that is shared by the processors directly and by all other nodes via the network. To achieve good scalability, it is essential that the caches be used effectively and that crucial data structures are *placed* in memories close to the processors that will use the structures. This placement is treated in Section 4.2

The implementation rests on two abstractions: a *Barrier* and a *Counter*. These abstractions are implemented in terms of IRIX threads (SPROCS) for the O2000 or POSIX threads for other platforms. A Barrier B has the expected semantics: when a thread calls $enter(B)$, it returns only after all other threads have called *enter*.

A Counter is a thread-safe counter that has two primary routines: $reset(C)$ and $next(C, p)$ (increment), where C is a Counter and p is a thread number. Counter is used to loop over groups at each level in the FMM. The *reset* routine sets the counter to zero and acts as a barrier. The *next* method returns the next value of the counter. The basic usage is that all the threads initialize the counter to zero with *reset* and then enter a loop getting the next value of the counter until all the groups at a given level have been processed.

There is one additional detail. At a given level in the FMM grouping there are a certain number of groups M_l. Assuming P threads, *next* for a thread p first returns values in the range $M_l p/P \ldots M_l(p+1)/P - 1$. These are the thread's groups for the level. Once a thread is done processing its groups, *next* begins to return values corresponding to groups the have not yet been processed by the other threads. When all work is complete, *next* returns a value $\geq M_l$ and the computation moves on to the next step. The net effect is a sort of dynamic load balancing. This is easy with shared memory, but difficult to achieve with explicit message passing. Two final Counter routines are $first(C, p)$ which returns $M_l p/P$ and $last(C, p)$ which gives $M_l(p+1)/P - 1$.

4 Parallel FMM

A basic parallel FMM is presented next that is implemented in terms of the primitives defined above. The basic algorithm is then modified to improve scalability by explicitly placing data structures in memory and by ordering the use of the translation operators.

4.1 Basic Algorithm

There are two routines: *setup* which builds the data structures necessary for the FMM, and *apply* which computes the product $Z \cdot I$.

The setup routine works as follows. First, a tree of groups is constructed. The lowest level ($l = 0$) groups contain elementary sources. Each higher level group at some level l, contains up to eight level $l - 1$ subgroups of one half the size. However, since a surface is being discretized, the typical number of subgroups is about four. The top of the tree consists of a single group which contains the entire scatterer. The quantity H is the height of the tree in levels, so that the topmost level is $H - 1$. Let $groups(l)$ be the set of groups at level l, and M_l be the number of elements in this set. Denote the parent of a group m by m_p. Finally, let L_l be the number of terms in the expansion at level l as determined by Equation 9.

For each group m two sets (lists) are constructed, $nearby(m)$ and $far(m)$, based on the following conditions:

$$m' \in nearby(m) \text{ iff } k_0 \mathbf{X}_{mm'} < L_l, \tag{11}$$

$$m' \in far(m) \text{ iff } m' \notin nearby(m) \text{ and } k_0 \mathbf{X}_{m_p m'_p} < L_{l+1} \tag{12}$$

where m and m' are members of *groups*(l), and $\mathbf{X}_{mm'}$ is the vector between the group centers $\mathbf{X}$ and $\mathbf{X}'$. In other words, a group is in the nearby list of m if it is too close to use the translation operators at that level. Otherwise, it is in the far list as long as the parents of m and m' are too close to use *their* translation operators. Interactions between sources are accounted for at the highest possible level.

The construction of the tree is fast and is done by the main thread. The main thread then creates P *apply threads* where P is typically set to the number of processors available. These threads perform memory allocation and construct the translation operators μ as described in Section 4.2. Once the apply threads have finished initializing, the setup is complete, and the threads wait on a Barrier.

When the iterative solver needs to compute $B = Z \cdot I$ (i.e. apply the operator), it releases the threads from the Barrier and they execute the steps listed below. The steps are written in terms of top level loops over groups using Counters. This naturally splits the work over threads and, hence, processors. This approach scales properly given good placement of data structures and care in applying translation operators. These issues are treated in more detail in the next sections. In what follows, the $\beta(\hat{k})$ quantities are denoted by s and the $\alpha(\hat{k})$ quantities are denoted by g. Loops are written in a C-style as `for` $(initialization; test; update)$, or as `for` $(i \in set)$ where i is understood to sequentially take on all values of the set or range. Each thread p executes the following to carry out the FMM apply:

Local-to-Far: The far field basis of each $l = 0$ group is constructed from its sources. There is no need to compute the multipole coefficients since it is a simple matter to compute the far-field directly from the sources.

```
for $(reset(C_0); m < M_0; m = next(C_0, p))$
    for $(k \in 0 \ldots K_0 - 1)$
        $s_{mk} = \sum_{a \in sources(m)} \lambda(\hat{k}, \mathbf{X}_m - \mathbf{x}_a) I_{ma}$
```

Note that at every level in the tree, there is a Counter C_l controlling the iterations at that level. The number of far field directions at a level is $K_l = 2L_l^2$ using the quadrature rule described Section 2. It should be clear that each value of an index k represents some $\hat{k} = (k_\theta, k_\phi)$ in the discretized far field basis for that level. The sources of a $l = 0$ group m are *sources*(m), and the location of a source a is $\mathbf{x}_a$.

Uptree: The far fields due to each subgroup of a group are interpolated and shifted to the group's center and accumulated to form the far field basis of the parent group.

```
for $(l \in 1 \ldots H - 1)$
    for $(reset(C_l); m < M_l; m = next(C_l, p))$
        for $(m' \in subgroups(m))$
            $\tilde{s}_{m'} = interpolate(s_{m'})$
            for $(k \in 0 \ldots K_l - 1)$
                $s_{mk} = s_{mk} + \lambda(\hat{k}, \mathbf{X}_m - \mathbf{X}_{m'}) \tilde{s}_{m'k}$
```

Translate: For each group m, the far field of each far away group is translated to m, converted to a j-expansion, and accumulated. This gives the field due to all groups far from m as a j-expansion valid inside of m.

```
for $(l \in 0 \ldots H-1)$
    for $(m \in first(C_l, p) \ldots last(C_l, p))$
        for $(m' \in far(m))$
            for $(k \in 0..K_l - 1)$
                $g_{mk} = g_{mk} + \mu(\hat{k}, \mathbf{X}_m - \mathbf{X}_{m'}) s_{m'k}$
```

Downtree: The j-expansions are walked down the tree in a way analogous to Uptree. The code works downward from level $H-1$, shifting the field g_m of group m to its subgroups and then filtering (instead of interpolating). The parallel structure is just like Uptree.

Far-to-Local: At the bottom of the tree, the j-expansions are used to evaluate the field at each source due to all far away sources. The procedure is the same as the Local-to-Far step except that $\hat{k} \rightarrow -\hat{k}$. At the end of this step, the result (B) has been computed for all far away interactions.

Direct: To account for interactions between groups that are too close to each other to use the FMM, the Green function is used directly:

```
for $(reset(C_0); m < M_0; m = next(C_0, p))$
    for $(m' \in near(m))$
        for $(a \in sources(m))$
            $B_{ma} = B_{ma} + \sum_{a' \in sources(m')} G(\mathbf{x}_a - \mathbf{x}_{a'}) I_{a'}$
$enter(apply\_gate)$
```

G is the Helmholtz kernel as defined in Equation 1. The final step is for all of the treads to enter a barrier. This ensures that the calculation is complete before returning to the main thread.

This description of the parallel algorithm is very similar to its sequential counterpart. The only complications are operations on the Counters, which look like regular loops, and the Barriers. These similarities between the parallel and sequential algorithm make implementation and maintainability easier.

This algorithm is for scalar (acoustic with Dirichlet boundary conditions) scattering. For the vector case (electromagnetic), the work doubles because two field components must be kept for each source but the algorithm is otherwise straightforward. The results in Section 5 are for electromagnetic scattering.

4.2 Memory Allocation and Placement

To assist in placing data structures in memory, IRIX provides an interface called `dplace`. During initialization, `dplace` is instructed to reserve $P/2$ local memories in a cube architecture. When each thread p is created during the FMM setup phase, it instructs `dplace` to associate itself with memory $p/2$. The default memory allocation policy in IRIX is "first-touch," meaning that when a thread allocates memory, IRIX attempts to satisfy the request on the node containing the processor that is currently executing the thread. The net effect, is that all memory allocated by an apply thread will be local assuming that the allocations can fit in its node.

In what follows, the phrase that a node allocates memory, indicates that one of the threads running on the node (like the even numbered thread), allocates the memory and then the other thread on the node aliases the allocation. This allows certain read-only data structures to be replicated across nodes but shared by the threads running on the node.

After the memory model is set up using `dplace`, a set of filters for moving between the different levels in the tree are allocated on each node. The filters at the lower several levels of an FMM tree are based on moderate sized matrices. Without local filters, Uptree and Downtree do not scale properly because there is a bottleneck when all processors try to fetch the matrices out of a single node. Similarly, the shift operators λ are replicated in each node since there are at most eight per level. (Except for the $l = 0$ shift operators, which are computed as needed.)

Each thread allocates the field variables s and g for its groups as well as local thread temporary storage (and working storage for the FTT routines used by the filters). In addition, every thread allocates and computes its share of translation operators (μ) that are used by all threads. Replication of the translation operators is unfeasible due to their size. This will have implications which are treated in Section 4.3.

The end result is that each node contains filters (and interpolators), shift operators, group field variables s and g, thread local storage, and a share of the translation operators. All of the other data structures required for the FMM, and there are many, are allocated without concern for placement because they are not performance critical.

4.3 Application of Translation Operators

Applying the translation operators in a scalable way is more problematic. Here the fields of all far away groups from a particular group are translated, converted to a j-expansion valid inside the group, and summed. It is likely that the field of a far away group will be in a remote node which makes this step highly cache sensitive. If naively implemented, the application of translation operators scales very poorly. Developing a method so that remote fields (fields of far away groups that are stored in remote nodes) are brought into the local cache and reused several times is essential to the overall scaling of the algorithm.

A simple observation is the key to scalability. Consider several groups that are neighbors, i.e. close together in space. If one of these groups needs a particular remote field, it is likely that its neighbors will also need the field since the distances between the neighbors and the remote group are roughly the same. The essential idea is to translate the remote field to all of the neighbors in succession which brings the field into the cache and reuses it many times.

To implement this idea, we need a ordering (numbering) of the groups for each level in the tree that keeps groups that are close together in space also close together in the ordering. Such an ordering is given by a breadth-first traversal of the group tree. A breadth-first traversal at a level is defined as follows. For the top-most level $H - 1$ the traversal is just to visit the single top-most group. To traverse level $l < H - 1$, visit all of the groups which are at level $l + 1$ in breadth-first order and for each level $l + 1$ (parent) group visit each of its subgroups. Since the subgroups are contained within the region of the parent, we get an ordering that keeps groups close together in space. This ordering is analogous to the Morton order reported in [10].

One final issue has to do with the small size of the cache. The basic loop for applying translation operators applies all operators to a group m before moving on to the next group in the ordering. It must be done this way in order to keep g_m (the far field representation

Processors	Time (s)	Speedup	Efficiency (%)
1	607.9	1	100
2	298.4	2.0	100
4	152.3	4.0	100
8	79.6	7.6	96
16	42.6	14.3	89
32	23.6	25.9	81

TABLE 1

Scalability of threaded multilevel FMM.

of the j-expansion for the group) in the cache as well. Caches are too small, however, to keep all of the remote fields at once, defeating the purpose of the ordering. The solution is to translate only a piece of the far field representation of a far away group at a time. The specific size of the pieces depends primarily on the cache size, but limiting the piece size kps to about $kps = 80$ double precision complex numbers has worked well in practice on several machines. So, at a given level, the ordering is traversed translating a piece of the far field representation for each group. At the end of the ordering, the process moves on to the next piece of the representation. This is repeated until all the far fields have been translated at that level. The code then continues onto the next level. The algorithm is very cache friendly.

In detail, translate is implemented as follows:

```
for ($l \in 0 \ldots H-1$)
  for ($kk = 0; kk < K_l; kk = kk + kps$)
    $ksize = min(kslice, K_l - kk)$
    for ($m \in first(C_l, p) \ldots last(C_l, p)$)
      for ($m' \in far(m)$)
        for ($k \in kk \ldots kk + ksize - 1$)
          $g_{mk} = g_{mk} + T_{mm'k} s_{m'k}$
```

where $T_{mm'k} = \mu(\hat{k}, \mathbf{X}_m - \mathbf{X}_{m'})$. These are the quantities that are precomputed in the setup phase. The effectiveness of the new implementation is demonstrated in the next section.

5 Results

The scaling of the threaded multilevel FMM apply algorithm is given in Table 1. Listed is the apply time in seconds versus the number of processors for a 16λ radius sphere discretized by 153,600 unknowns. Also listed is the speedup $S_p = T_1/T_p$ where T_p is the apply time for p processors, and the parallel efficiency $100 S_p/p$. The scaling is very good, with 32 processors achieving 81% efficiency.

The effect of the technique used to apply the translation operators is shown in Table 2 for the same problem. The table shows the total time spent by all processors in the Translate step. Using the technique described in Section 4.3 , the effort to apply the operators grows by 29.3% as the number of processors increases from 1 to 32 (the elapsed time is 82.5s for 1 processor and 3.33s for 32 processors). In contrast, if the operators are applied naively without ordering the groups or dividing up the far field directions for cache efficiency, the effort to apply the operators grows 173% and begins to take a substantial fraction of the total apply time.

The scaling of the apply can be further improved by additional tuning in Uptree and Downtree. The main problem is that static data for the filters is not replicated across the

Processors	1	2	4	8	16	32
Scalable (s)	82.5	81.7	83.7	88.2	94.5	106.7
Unscalable (s)	99.1	110.4	124.5	158.8	204.9	271.0

TABLE 2

Time spent doing translations versus number of processors for scalable and unscalable implementations.

nodes which causes a bottleneck (filter dynamic data. like the matrices, *are* replicated). This can be improved with some programming effort.

6 Concluding Remarks

The threaded approach taken here has some advantages over explicit message passing. Often some of the interprocessor communications required in complex parallel codes are *not* performance sensitive. Such communications can be handled automatically by the hardware in a threaded shared memory approach without burdening the programmer. Making the performance sensitive parts work properly, i.e. scale, is largely an exercise in tuning the caches which must be done regardless for good uniprocessor performance.

In addition, there is a maintainability benefit. As fast scattering codes gets more complicated, with the addition of support for complex materials and subwavelength structures, the load balancing problem implicit in message passing codes will become very complex. Parallelizing such codes will be easier in a shared memory environment.

Significantly, the compact size of the FMM allows the exploitation of another form of parallelism: computing the scattering from multiple incident angles. With large $O(N^2)$ operators the entire machine would be needed just to store the operator. The FMM is far more compact and can be replicated several times in a supercomputer, making the multiple angle problem embarrassingly parallel. The same is true for design optimization (parameter) studies.

The parallel FMM presented here is part of the FastScat program for performing electromagnetic scattering calculations. Recently, FastScat computed the radar cross section for both polarizations of an 40λ radius sphere to 0.16 db rms accuracy in 20.7 hours on a 32 node Origin 2000[1]. The target was over $20,000$ square wavelengths. The ability to accurately compute the RCS of such a large target is due to the FMM, a discretization of the integral equation that is of high order[1], and a scalable parallel implementation of the FMM.

Acknowledgements

The results presented here were from runs at the Army Research Laboratory Major Shared Resource Center. I would like to thank John Visher of HRL Laboratories for implementation assistance and Tom Kendall of ARL MSRC for support in using the machines.

References

[1] L.F. Canino, J.J. Ottusch, M.A. Stalzer, J.L. Visher, and S.M. Wandzura, *Numerical solution of the Helmholtz equation in 2d and 3d using a high-order Nyström discretization,* J. Computational Physics, 146, 1998, pp. 627–663.

---

[1] This calculation was performed with an earlier version of the FMM code described here.

[2] R. Coifman, V. Rokhlin, and S. Wandzura, *The fast multipole method: a pedestrian prescription,* IEEE Antennas and Propagation Mag., 3 (35), June 1993, pp. 7–12.

[3] M.A. Epton and B. Dembart, *Multipole translation theory for the three-dimensional Laplace and Helmholtz equations,* SIAM J. Scientific Computing, 4 (16), July 1995, pp. 865–897.

[4] L. Greengard and W.D. Grop, *A parallel version of the fast multipole method,* Computers Math. Applic., 20 (1990), pp. 63–71.

[5] M.F. Gyure and M.A. Stalzer, *A prescription for the multilevel Helmholtz FMM,* IEEE Computational Science & Engineering, July-Sept. 1998, pp. 39–47.

[6] V. Rokhlin, *Diagonal form of translation operators for the Helmholtz equation in three dimensions,* Applied and Computational Harmonic Analysis, 1(1), Dec. 1993, pp. 82–93.

[7] V. Rokhlin and M.A. Stalzer, *Scalability of the fast multipole method for the Helmholtz equation,* Proc. Eighth SIAM Conf. on Parallel Processing for Scientific Computing, March 1997, Minneapolis, MN.

[8] J.P. Singh, C. Holt, J.L. Hennessy, and A. Gupta, *A parallel adaptive fast multipole method,* Proc. Supercomputing '93, Nov., Portland, OR, pp. 54–65.

[9] M.A. Stalzer, *A parallel fast multipole method for the Helmholtz equation,* Parallel Processing Letters, 2 (5), 1995, pp. 263–274.

[10] M.S. Warren and J.K. Salmon, *A parallel, portable and versatile treecode,* Proc. Seventh SIAM Conf. on Parallel Processing for Scientific Computing, Feb. 1995, San Francisco, CA.

[11] N. Yarvin and V. Rokhlin, *A generalized 1D fast multipole method, with applications to filtering of spherical harmonics,* tech. report, 1999, Dept. of Computer Science, Yale University, New Haven, CT.

Scalable Electromagnetic Scattering Calculations
on the
SGI Origin 2000*

John J. Ottusch Mark A. Stalzer John L. Visher
Stephen M. Wandzura

Information Sciences Laboratory
HRL Laboratories, Malibu, California

March 24, 2000

Abstract

We describe the FastScat™ program for electromagnetic scattering calculations and its parallel implementation on the SGI Origin 2000. FastScat recently computed the radar cross section of a sphere having an area of $45,239\lambda^2$ to high accuracy in about a day. This is contrasted with a result for an $354\lambda^2$ sphere reported at Supercomputing '92. Taking both size and accuracy into account, the FastScat result represents an improvement in solution time of over nine orders of magnitude. This improvement was due to systematically focusing on several issues that impact the scalability of electromagnetic scattering calculations.

1 Introduction

This paper presents the FastScat™ program for efficiently performing frequency domain electromagnetic scattering calculations using a boundary integral equation formulation on parallel computers. Typical applications include radar cross section (RCS) prediction, the computation of antenna radiation patterns, and high-frequency circuit package modeling. FastScat is a truly *scalable* code in that:

- additional accuracy in a computed solution can be achieved at low cost;
- a small increase in problem size (area) causes only a modest increase in computer resources; and
- the code shows good parallel scalability.

The scalability of FastScat allows us to perform scattering calculations for very large objects. As an example, FastScat recently computed the RCS of a metal sphere having an area of $45,239\lambda^2$ (radius $r = 60\lambda$) to high accuracy in about a day. This is in contrast to the result for an $354\lambda^2$ sphere computed by the Patch code running on the Intel Touchstone Delta reported at Supercomputing '92[3]. Taking both size and accuracy into account, the FastScat result represents an improvement in solution time of over nine orders of magnitude.

Scattering cross sections and radiation patterns can be computed by solving a matrix equation, $Z \cdot I = V$, derived from the discretization of an integral equation. The number of unknowns N required for accurate modeling of such problems can be very large, which can severely limit problem size. The system can be solved by factoring the dense matrix Z (using $O(N^3)$ operations), or by using an iterative method which requires $O(N^2)$ operations per iteration. Each iteration of an iterative solver involves the multiplication of

---

*This work was supported by the Defense Advanced Research Projects Agency, the Air Force Office of Scientific Research, Hughes Electronics, and the Raytheon Systems Company. Computer runs were performed at the Army Research Laboratory's Major Shared Resource Center in Maryland.

an approximate solution I for the source distribution by the impedance matrix Z. The iteration count must be controlled to achieve reasonable solution times.

There are four important solution method characteristics required to achieve scalabilty:

- *The method must be high order.* It is desirable that computed solutions converge as h, the characteristic scale size of the discretization, decreases. For boundary integral solutions to scattering problems, the error ϵ generally scales as $\epsilon \propto h^p$, where $p \geq 1$ is the order of convergence. Most codes based on the Method of Moments, such as Patch, are low order: $\epsilon \propto h^2$. In contrast, FastScat is high order and values of p up 10 are routinely used. High order convergence allows us to get extra accuracy for minimal additional computational cost. This is essential for estimating the solution error and computing scattering from objects with large dynamic ranges[1].
- *The method must be fast.* An $O(N^3)$ method is feasible only for small problems. By using an iterative solver and switching to the Fast Multipole Method (FMM)[2, 4, 6, 7, 9], the time complexity can be reduced to $O(C_i N \log^2 N)$ where C_i is the iteration count. The FMM constructs a sparse representation of Z which is used to efficiently compute the product $Z \cdot I$.
- *The integral equation must be well conditioned.* FastScat uses a Combined Field Integral Equation formulation (CFIE)[9] which results in a well conditioned operator for many scatterers. The CFIE, in conjunction with a simple preconditioner and a conjugate gradient solver, keeps the iteration count C_i reasonable.
- *The implementation must have good parallel scalability.* The crucial parallel operation in FastScat is applying the FMM. All other computations are either embarrassingly parallel or are so cheap that they can be done on a single processor. A substantial amount of work has been done on parallelizing the FMM for the Laplace equation[5, 8, 13]. For electromagnetic scattering, the Helmholtz FMM is required. Achieving good parallel scalability with this variant of the FMM poses some additional challenges[10, 11].

Some parts of this work have been previously reported[1, 6, 9, 11]. Here we show how all of the parts fit together to enable the solution of very large scattering problems. In total, we believe this work serves as the current benchmark for the state of the art in frequency domain electromagnetic scattering calculations.

This paper is organized into a section on each aspect of scalability, followed by a results section and some concluding remarks.

2 Discretizating the Integral Equation

Here we consider a prototypical scattering problem — $3d$ scalar scattering with Dirichlet boundary conditions — to show how the linear system $V = Z \cdot I$ is formed. This will set the stage for the following sections on discretizations and the FMM.

A specified field $\phi(\mathbf{x})$ on a surface S induces an unknown source distribution $\sigma(\mathbf{x}')$ on S. This distribution radiates a scattered field

$$\psi(\mathbf{x}) = \int_S G(\mathbf{x} - \mathbf{x}')\sigma(\mathbf{x}')d\mathbf{x}' \tag{1}$$

where the Green function is

$$G(r) = \frac{e^{ik_0 r}}{r} \tag{2}$$

k_0 is the wave number ($k_0 = 2\pi$ in free space for dimensions in wavelengths), and $r = |\mathbf{x} - \mathbf{x}'|$. Applying the Dirichlet boundary condition $\phi(\mathbf{x}) + \psi(\mathbf{x}) = 0$ for $\mathbf{x}$ on S, gives

$$\phi(\mathbf{x}) = -\int_S G(\mathbf{x} - \mathbf{x}')\sigma(\mathbf{x}')d\mathbf{x}', \quad \mathbf{x} \text{ on } S. \tag{3}$$

For a moment, ignore the singular nature of G. This integral can be evaluated numerically by choosing a suitable N-point quadrature rule. Evaluating Equation 3 at the ith abscissa of the quadrature rule gives

$$V_i = -\sum_{j=1}^{N} w_j G_{ij} I_j \tag{4}$$

where $V_i = \phi(\mathbf{x}_i), G_{ij} = G(\mathbf{x}_i - \mathbf{x}_j)$, and w_j is the weight of the jth sample point (at $\mathbf{x}_j$) of the quadrature rule. We want to solve this linear system for the unknown sources I. From I, we can easily compute the scattered field at any place exterior to S.

Equations equivalent to Equation 3 are also available for electromagnetic scattering. FastScat uses the Combined Field Integral Equation formulation which is well conditioned and immune to spurious internal resonances[9]. Using the CFIE in conjunction with a simple preconditioner[1] and a conjugate gradient type solver keeps iteration counts reasonable. For the $r = 60\lambda$ sphere, only 19 iterations were required for roughly two digits of accuracy.

3 High Order Discretizations

The quadrature rule used in Equation 4 is selected so that it integrates a certain class $\mathcal{F}$ of functions over S exactly. If the source distribution can be represented exactly as an expansion over $\mathcal{F}$ then the convolution can be computed exactly.

In practice, the source distribution on an arbitrarily-shaped surface can be well approximated by dividing it into patches and locating the sample points on each patch according to a quadrature rule that can integrate polynomials exactly up to order p. In the case of quadrilaterals, an appropriate rule is formed from the product of two Gauss-Legendre rules. Analogous rules exist for triangles[12]. The overall discretization will converge with $O(h^p)$ assuming expansions over $\mathcal{F}$ are accurate to that order.

This works extremely well for regular kernels, but Nature is not so kind and the Helmholtz kernel G behaves poorly as the points i and j become close. When this happens, the quadrature rule needs to be adjusted to account for the singular and oscillatory nature of G. The proper adjustment is achieved by replacing the discretized Green function in Equation 4 by

$$G_{ij} = \begin{cases} G(\mathbf{x}_i - \mathbf{x}_j) & \text{if } \mathbf{x}_i \text{ is far from } \mathbf{x}_j \\ L_{ij} & \text{otherwise} \end{cases} \tag{5}$$

where the L_{ij} are known as the "local corrections"[1]. The definition of "far from" depends on the desired accuracy. In practice it is about a half wavelength for two digits.

For a given field point i, the L_{ij} are computed by solving the linear system

$$\sum_j w_j L_{ij} f^{(k)}(\mathbf{x}_i - \mathbf{x}_j) = \int_{D_i} G(\mathbf{x}_i - \mathbf{x}') f^{(k)}(\mathbf{x}_i - \mathbf{x}') d\mathbf{x}' \tag{6}$$

for all the testing functions $f^{(k)}$ in $\mathcal{F}$. The region D_i is the local domain of the ith field point. This region is determined by computing the right hand side of Equation 6 adaptively, on a patch by patch basis, and comparing it to the left hand side quadrature. This procedure proceeds until the difference is below some error tolerance. The local corrections L_{ij} for points outside of D_i are zero so the linear system is small. The number of points in D_i may be different from the number of testing functions in $\mathcal{F}$, in which case, singular value decomposition is used to solve the system. However, it is often possible to arrange the system so that the number of points and functions are the same. This approach restores the desired order of convergence, which has been shown on many scatterers.

In terms of scalable scattering calculations, high order discretizations allow us to check the accuracy of solutions relatively cheaply. They also allow us to often compute a solution to a given accuracy with fewer unknowns.

4 Fast Multipole Method

The FMM computes $B_i = \sum_j w_j G_{ij} I_j$ (Equation 4) for all points i in $O(N \log^2 N)$ time. This is the product $Z \cdot I$ needed by the iterative solver. This section presents the basics of the Helmholtz FMM.

---

[1] The preconditioner is block diagonal and represents the inverse of some FMM group self-interactions. It works well for many scatterers, but does not remove all of the ill-conditioning in the formulation. Generalizations to the CFIE are currently being explored and some look very promising.

Consider two well-separated spheres of radius R_1 and R_2, each containing a collection of Helmholtz sources. We want to quickly evaluate the field generated by all the sources in R_1 at every source in R_2. This field can be written as a multipole expansion valid outside of R_1 as

$$\psi(\mathbf{r}) = \sum_{lm} \beta_{lm} h_l(kr) Y_{lm}(\theta, \phi) \tag{7}$$

where r, θ, and ϕ are relative to a coordinate system centered in R_1, $h_l(kr)$ are spherical Hankel functions of the first kind, and $Y_{lm}(\theta, \phi)$ are normalized spherical harmonics. We refer to this expansion as an h-expansion. Similarly, we can write an expression for the field valid inside R_2 :

$$\varphi(\mathbf{r}) = \sum_{lm} \alpha_{lm} j_l(kr) Y_{lm}(\theta, \phi) \tag{8}$$

where the coordinate system is now centered in R_2, and $j_l(kr)$ are spherical Bessel functions. We refer to this expansion as a j-expansion. For the moment, we consider both of these to be infinite sums. The FMM then rests on three observations:

- The origin of an h-expansion can be shifted arbitrarily inside R_1, and a new set of coefficients, $\tilde{\beta}_{lm}$, can be computed for this new expansion. The same holds for shifting a j-expansion arbitrarily to a new origin inside of R_2, which results in a new set of coefficients, $\tilde{\alpha}_{lm}$.
- An h-expansion valid outside of R_1 can be translated and converted into a j-expansion valid inside R_2.
- Most crucial, these shifts and translations can be done efficiently by transforming the coefficients into a basis in which both operators are diagonal.

The *far-field* transform of an arbitrary function $f(\hat{k})$ is

$$f(\hat{k}) = \sum_{lm} i^l Y_{lm}(\hat{k}) f_{lm} \tag{9}$$

and the inverse transform is given by

$$f_{lm} = \int d\hat{k} i^{-l} Y^{*}_{lm}(\hat{k}) f(\hat{k}) \tag{10}$$

where $\hat{k}$ is a unit vector represented by polar and azimuthal angular components (k_θ, k_ϕ).

It is in this k-basis that the shift and translation operators are diagonal. An h-expansion in its far-field basis is shifted from a point $\mathbf{x}$ to another point $\mathbf{x}'$ both inside of R_1 by

$$\tilde{\beta}(\hat{k}) = \lambda(\hat{k}, \mathbf{x}' - \mathbf{x}) \beta(\hat{k}) \tag{11}$$

where λ is given by

$$\lambda(\hat{k}, \mathbf{x}' - \mathbf{x}) = e^{ik_0 \hat{k} \cdot (\mathbf{x}' - \mathbf{x})} \tag{12}$$

The same shift operator λ also applies to j-expansions. It represents a "local" shift in the group center, retaining the exterior or interior expansion.

The translation of an h-expansion into a j-expansion is through the translation operator μ, which, in the far-field basis is

$$\mu(\hat{k}, \mathbf{x}' - \mathbf{x}) = \sum_{l} i^l (2l+1) h_l(k_0 |\mathbf{x}' - \mathbf{x}|) P_l(\hat{k} \cdot (\mathbf{x}' - \mathbf{x}) / |\mathbf{x}' - \mathbf{x}|) \tag{13}$$

where the P_l are Legendre polynomials.

In practice the expansions are truncated to a finite number of terms L depending on the group size and desired accuracy. The mathematical validity of this truncation is addressed by Rokhlin[7] but it is related to the fact that these series are asymptotic and are, therefore, of controllable accuracy. Empirically, it has been determined that the number of terms L needed in the expansions for a region of diameter D is[2]

$$L = k_0 D + \frac{d}{1.6} \log(k_0 D + \pi) \tag{14}$$

where d is the desired number of digits.

The above expressions for the translation operators, together with the far-field transform, are the basic tools used to construct a multilevel FMM algorithm. Clearly the field caused by a collection of sources inside an arbitrary group G_1 can be evaluated at any point inside a second group G_2 by converting the exterior h-expansion, valid outside G_1, to an interior j-expansion which is valid inside G_2. Also, we can calculate the field at that point caused by the sources in G_1 by computing $\tilde{\alpha}_{00}$, the leading term in the j-expansion. No other terms contribute, because the expansion is already centered at the field point where $r = 0$ and all the terms $j_l(0)$ are zero except for j_0 which is one. Thus, we can evaluate the field directly through the far-field transform as

$$\phi(0) = \tilde{\alpha}_{00} = \frac{1}{\sqrt{4\pi}} \int d\hat{k}\tilde{\alpha}(\hat{k}). \qquad (15)$$

The abcissae $\hat{k} = (k_\theta, k_\phi)$ of the numerical quadrature rule used to compute this integral are selected so that it can be performed exactly. One choice is to use a trapezoidal rule of $2L$ points in the ϕ direction and an L point Gauss-Legendre rule in the θ direction. This discretization of the $\hat{k}$ basis is used throughout the FMM.

The multilevel Helmholtz FMM works in fundamentally the same way as the Laplace FMM in that it combines expansions valid inside the original groups to form expansions valid inside correspondingly larger groups with bigger group diameters. This recursive regrouping results in a tree-like structure that has groups of different sizes at different levels of the tree. The h-expansions from neighboring groups are shifted and combined into a single h-expansion representing a larger group when going up the tree, and j-expansions in a large group are converted to smaller groups going down the tree. The details of this process are given in Section 5.1.

There is, however, an important mathematical detail. When going up the tree, it is necessary to interpolate the far-field representation of a group at one level onto the denser ($\hat{k}$ more closely spaced) basis of the group one level higher. Similarly, when going down the tree, it is necessary to convert to a sparser basis in a filtering process. In both cases, the code converts from the far-field basis to the multipole coefficients and then back to the new far-field basis using the definitions given in Equations 9 and 10. The actual implementation is in terms of fast Fourier transforms for the k_ϕ direction, and fast associated Legendre transforms for the k_θ direction[14]. As a practical matter, a slow associated Legendre transform which is implemented in terms of matrix multiplication can be used on rather large problems because of the small prefactor in its time complexity relative to the fast transform. However, fetching the transform matrices from memory causes some scalability problems which are addressed in Section 5.2. The details of the filtering and interpolation processes are given in [6].

5 Parallel Implementation

FastScat is implemented in a threaded style assuming a cache-coherent distributed shared memory machine. On the O2000, it uses IRIX threads (SPROCs)[11]. A POSIX threads version is also available. It order to achieve parallel scalability, it is essential that the local processor caches be used effectively and that selected data structures are replicated to reduce network contention.

A FastScat run progresses through three phases: setup, solve, and RCS computation. The setup computes the local corrections L_{ij}, and is embarrassingly parallel. The scalability is good to about 32 processors and then begins to fall off due to contention over the discretization data structures. The RCS computations are also easy to parallelize. Perfect scalability in the setup and RCS phase are not presently a concern since, on practical problems, FastScat spends most of its time solving for the surface currents for various excitations ("look angles")[2].

The solve phase uses the iterative solver, preconditioner, and FMM. The preconditioner can be applied in parallel easily (backsubstitution of the blocks), and the iterative solver does inner products over relatively short vectors (at most a few million elements) which can be done on a single processor. Naive implementations of the FMM, however, scale very poorly. On the O2000, there is hardly any benefit to using more than a few processors. The remainder of this section describes the implementation of FastScat's parallel FMM.

---

[2] The sphere run spends more of its time in setup since there is only one look angle.

5.1 Parallel FMM

There are two primary FMM routines: *setup* which builds the data structures, and *apply* which computes the product $Z \cdot I$.

The setup routine works as follows. First, a tree of groups is constructed. The lowest level ($l = 0$) groups contain elementary sources. Each higher level group at some level l, contains up to eight level $l - 1$ subgroups of one half the size in each linear dimension. However, since a surface is being discretized, the typical number of subgroups is about four. The top of the tree consists of a single group which contains the entire scatterer. The quantity H is the height of the tree in levels, and the topmost level is $H - 1$. Let $groups(l)$ be the set of groups at level l, and M_l be the number of elements in this set. Denote the parent of a group m by m_p. Finally, let L_l be the number of terms in the expansion at level l as determined by Equation 14.

For each group m two sets (lists) are constructed, $nearby(m)$ and $far(m)$, based on the following conditions:

$$m' \in nearby(m) \quad \text{iff} \quad k_0 \mathbf{X}_{mm'} < L_l, \tag{16}$$

$$m' \in far(m) \quad \text{iff} \quad m' \notin nearby(m) \text{ and } k_0 \mathbf{X}_{m_p m'_p} < L_{l+1} \tag{17}$$

where m and m' are members of $groups(l)$, and $\mathbf{X}_{mm'}$ is the vector between the group centers $\mathbf{X}_m$ and $\mathbf{X}_{m'}$. In other words, a group is in the nearby list of m if it is too close to use the translation operators at that level. Otherwise, it is in the far list as long as the parents of m and m' are too close to use *their* translation operators. Interactions between sources are accounted for at the highest possible level.

Once the tree is constructed, various quantities, such as the translation operators are computed. The setup routine is called only once.

When the iterative solver needs to compute $B = Z \cdot I$, it calls the apply routine. For most problems, FastScat spends most of its time in apply. Apply is implemented in terms of P threads where P is the number of processors. The apply steps are written in terms of loops over groups and it is a simple matter to split these loops over the threads. These loops are controlled by a thread-safe counter that has two primary routines: $reset(C)$ and $next(C, p)$, where C is a counter and p is a thread number. The $reset$ routine sets the counter to zero and acts as a barrier. The $next$ routine returns the next value of the counter. The basic usage is that all the threads initialize the counter to zero with $reset$, and then enter a loop getting the next value of the counter until all the groups at a given level have been processed. In addition, there are two routines $first(C, p)$ and $last(C, p)$ which together define a sequence of groups $first(C, p) \ldots last(C, p)$ that thread p can process efficiently because the data structures for the groups have been allocated locally (see Section 5.2).

To compute $B = Z \cdot I$, each thread p does the following:

Local-to-Far: The far-field basis of each $l = 0$ group is constructed from its sources. There is no need to compute the multipole coefficients since it is a simple matter to compute the far field directly from the sources.

```
for ($reset(C_0); m < M_0; m = next(C_0, p)$)
    for ($k \in 0 \ldots K_0 - 1$)
        $s_{mk} = \sum_{a \in sources(m)} \lambda(\hat{k}, \mathbf{X}_m - \mathbf{x}_a) I_{ma}$
```

Note that at every level in the tree, there is a counter C_l controlling the iterations at that level. The number of far field directions at a level is $K_l = 2L_l^2$ using the quadrature rule described in Section 4. It should be clear that each value of an index k represents some $\hat{k} = (k_\theta, k_\phi)$ in the discretized far field basis for that level. The sources of a $l = 0$ group m are $sources(m)$, and the location of a source a is $\mathbf{x}_a$. The vector s is simply the $\beta(\hat{k})$ quantities of Section 4.

Uptree: The far fields due to each subgroup of a group are interpolated and shifted to the group's center and accumulated to form the far field basis of the parent group.

```
for ($l \in 1 \ldots H - 1$)
    for ($reset(C_l); m < M_l; m = next(C_l, p)$)
        for ($m' \in subgroups(m)$)
            $\bar{s}_{m'} = interpolate(s_{m'})$
```

```
for ($k \in 0 \ldots K_l - 1$)
    $s_{mk} = s_{mk} + \lambda(\hat{k}, \mathbf{X}_m - \mathbf{X}_{m'})\tilde{s}_{m'k}$
```

Translate: For each group m, the far field of each far away group is translated to m, converted to a j-expansion, and accumulated. This gives the field due to all groups far from m as a j-expansion valid inside of m.

```
for ($l \in 0 \ldots H - 1$)
    for ($m \in first(C_l, p) \ldots last(C_l, p)$)
        for ($m' \in far(m)$)
            for ($k \in 0..K_l - 1$)
                $g_{mk} = g_{mk} + \mu(\hat{k}, \mathbf{X}_m - \mathbf{X}_{m'})s_{m'k}$
```

The vector g contains the $\alpha(\hat{k})$ quantities.

Downtree: The j-expansions are walked down the tree in a way analogous to Uptree. The code works downward from level $H - 1$, shifting the field g_m of group m to its subgroups and then filtering (instead of interpolating). The parallel structure is just like Uptree.

Far-to-Local: At the bottom of the tree, the j-expansions are used to evaluate the field at each source due to all far away sources. The procedure is the same as the Local-to-Far step except that $\hat{k} \rightarrow -\hat{k}$. At the end of this step, the result (B) has been computed for all far away interactions.

Direct: To account for interactions between groups that are too close to each other to use the FMM, the locally corrected kernel (Equation 5) is used directly:

```
for ($reset(C_0); m < M_0; m = next(C_0, p)$)
    for ($m' \in near(m)$)
        for ($a \in sources(m)$)
            $B_{ma} = B_{ma} + \sum_{a' \in sources(m')} G(\mathbf{x}_a - \mathbf{x}_{a'})I_{a'}$
```

This description of the parallel algorithm is very similar to its sequential counterpart. The only complications are operations on the counters, which look like regular loops. The similarities between the parallel and sequential algorithm make implementation and maintenance easier.

This algorithm is for scalar (acoustic with Dirichlet boundary conditions) scattering. For the vector case (electromagnetic), the work doubles because two field components must be kept for each source but the algorithm is otherwise straightforward. The results in Section 6 are for electromagnetic scattering.

5.2 Data Placement

For most FMM steps, memory references tend to be localized to the data associated with a particular group and its subgroups. In order to make these references efficient (accesses to local memory) each apply thread p is assigned a sequence of groups $first(C_l, p) \ldots last(C_l, p)$ at each level l. For example, if there are eight groups at a level and two threads, the first thread gets groups $1 \ldots 4$ and the second thread gets $5 \ldots 8$. As part of its initialization, each thread allocates certain key data structures, such as s and g for its sequence of groups. These allocations will generally go to the local memory since a first-touch memory allocation policy is used. Threads also set their processor affinities so that they are not moved away from their data structures by the operating system. One additional point is that counter's $next(C_l, p)$ routine first returns groups in thread p's sequence. Once the sequence is exhausted, it returns groups in the sequences of threads that are lagging behind in the computation. This acts as a form of dynamic load balancing[11].

A modest amount of data replication is also required. The routines *interpolate* and *filter* used by Uptree and Downtree contain several moderately sized matrices used in the filtering and interpolation process (Section 4). These must be replicated a few times to reduce network contention and preserve the scalability of Uptree and Downtree. Presently, FastScat replicates the matrices in every node (two processors), but this is probably an overkill.

Processors	Time (s)	Speedup	Efficiency (%)
1	607.9	1	100
2	298.4	2.0	100
4	152.3	4.0	100
8	79.6	7.6	96
16	42.6	14.3	89
32	23.6	25.9	81

TABLE 1

Scalability of threaded multilevel FMM for a $r = 16\lambda$ *sphere.*

5.3 Scalable Application of Translation Operators

Applying the translation operators in a scalable way is more problematic. Here the fields of all far away groups from a particular group are translated, converted to a j-expansion valid inside the group, and summed. It is likely that the field of a far away group will be in a remote node which makes this step highly cache sensitive. If naively implemented, the application of translation operators scales very poorly. Developing a method so that remote fields (fields of far away groups that are stored in remote nodes) are brought into the local cache and reused several times is essential to the overall scaling of the algorithm.

A simple observation is the key to scalability. Consider several groups that are neighbors, i.e. close together in space. If one of these groups needs a particular remote field, it is likely that its neighbors will also need the field since the distances between the neighbors and the remote group are roughly the same. The essential idea is to translate the remote field to all of the neighbors in succession which brings the field into the cache and reuses it many times. To do this, we need a ordering (numbering) of the groups for each level in the tree that keeps groups that are close together in space also close together in the ordering. Such an ordering is given by a breadth-first traversal of the group tree. This is analogous to the Morton order reported in [13].

One final issue has to do with the small size of the cache. The basic loop for applying translation operators applies all operators to a group m before moving on to the next group in the ordering. It must be done this way in order to keep g_m (the far-field representation of the j-expansion for the group) in the cache as well. Caches are too small, however, to keep all of the remote fields at once, defeating the purpose of the ordering. The solution is to translate only a piece of the far-field representation of a far away group at a time. The specific size of the pieces depends primarily on the cache size, but using a piece size (kps) of 80 double precision complex numbers has worked well in practice on several machines. So, at a given level, the ordering is traversed translating a piece of the far-field representation for each group. At the end of the ordering, the process moves on to the next piece of the representation. This is repeated until all the far fields have been translated at that level. The code then continues onto the next level.

In detail, translate is implemented as follows:

```
for ($l \in 0 \ldots H-1$)
   for ($kk = 0; kk < K_l; kk = kk + kps$)
      $ksize = min(kps, K_l - kk)$
      for ($m \in first(C_l, p) \ldots last(C_l, p)$)
         for ($m' \in far(m)$)
            for ($k \in kk \ldots kk + ksize - 1$)
               $g_{mk} = g_{mk} + T_{mm'k} s_{m'k}$
```

where $T_{mm'k} = \mu(\hat{k}, \mathbf{X}_m - \mathbf{X}_{m'})$. These quantities are computed in the setup phase.

5.4 FMM Parallel Scalability Results

The scaling of the threaded multilevel FMM apply algorithm is shown in Table 1. The apply time in seconds versus the number of processors is given for a $r = 16\lambda$ sphere discretized by 153,600 unknowns. The speedup $S_p = T_1/T_p$ (where T_p is the apply time for p processors) and the parallel efficiency $100 S_p/p$ are also listed. The scaling is very good, with 32 processors achieving 81% efficiency.

Tuned and naive implementations of the translation operator application are compared in Table 2 for the same problem. The table shows the total time spent by all processors in the translate step. The effort to apply

Processors	1	2	4	8	16	32
Tuned (s)	82.5	81.7	83.7	88.2	94.5	106.7
Naive (s)	99.1	110.4	124.5	158.8	204.9	271.0

TABLE 2

Time spent doing translations versus number of processors for tuned and naive implementations.

Year	1992	1999
Code	Patch	FastScat
Computer	Touchstone Delta	Origin 2000
Processors	512	64
Radius (λ)	5.31	60
Area (λ^2)	354	45,239
Accuracy (db rms)	2 (est)	0.12
Unknowns	48,673	2,160,000
Memory (Gb)	38	45.5
Time (hrs)	19.6	27.9

TABLE 3

State of the Art: 1992 vs. 1999

the operators grows by 29.3% as the number of processors increases from 1 to 32 (the elapsed time is 82.5s for 1 processor and 3.33s for 32 processors). In contrast, if the operators are applied without ordering the groups or dividing up the far field representation for cache efficiency, the effort to apply the operators grows 173% and begins to take a substantial fraction of the total FMM time.

6 Electromagnetic Scattering Results

FastScat was used to compute the bistatic RCS of a $r = 60\lambda$ sphere for both polarizations on a 64 processor SGI Origin 2000. Table 3 shows information from the run including problem area, accuracy (as compared to the Mie series solution), number of unknowns, memory required, and run time. It is compared to the 1992 result from the Patch code on the Touchstone Delta. The FastScat run times by phase were 20.2 hours for setup (mostly computing local corrections), 7.66 hours for the solve (computation of surface currents using the FMM), and 1.04 hours to compute the bistatic RCS at 1,800 angles. Figure 1 plots the computed RCS versus the Mie series solution. The two curves are nearly identical.

The Patch code used a tuned out-of-core solver to factor Z. The solver was carefully constructed to overlap disk I/O, interprocessor communication, and computation, to achieve high performance. It sustained a rate of 10.35 Gflops, which was within a factor of 2 of the theoretical maximum rate of the Delta for the inner loop of the computation. The Patch/Delta result represented the largest reported scattering run to date in 1992.

It would take Patch/Delta some time to match the FastScat result in both size and accuracy. In order for Patch to achieve an accuracy of roughly 0.2dB, the number of unknowns would have to be increased by about a factor of 10 due to the $O(h^2)$ convergence rate of its discretization. The difference in area is over a factor of 100. Taken together, the unknown count must increase $\sim$ 1000 fold. Since the factorization process is $O(N^3)$, the run time can be expected to increase by roughly nine orders of magnitude.

We have used FastScat to compute the RCS of a variety of benchmark targets. Figure 2 shows the currents induced on the Dart, a standard test case, at 18 GHz with the incident radiation nose-on. At this frequency, the Dart is $4441\lambda^2$ in area and is discretized by 436,000 unknowns. Figure 3 show the monstatic RCS in both polarizations using an over-the-top scan. This scan goes from the back at -90 degrees to the tip at 90 degrees. By using convergence studies, which are relatively inexpensive with a high order discretization, the error has been estimated at approximately 0.1dB in the high RCS regions and roughly 2 dB in the stealthy regions (near the tip). FastScat required 8.3 Gb of memory, did the setup in 3.0 hours, and solved for each monostatic angle in an average of 17 minutes. A 32 processor Origin 2000 was used. At 436,000 unkowns, the 18 GHz Dart is too large for dense matrix techniques even on the biggest supercomputers.

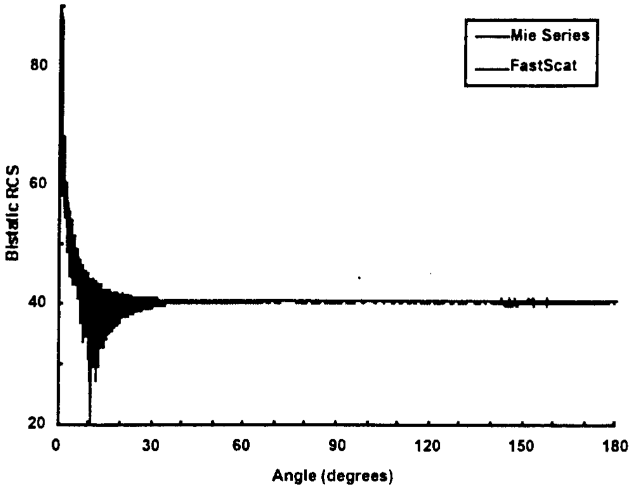


FIG. 1. *Computed RCS of a* $r = 60\lambda$ *sphere compared to the Mie series solution.*

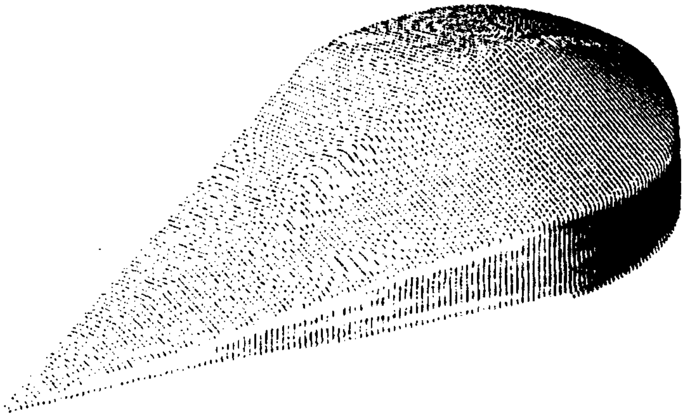


FIG. 2. *Computed surface currents of the Dart at 18 GHz.*

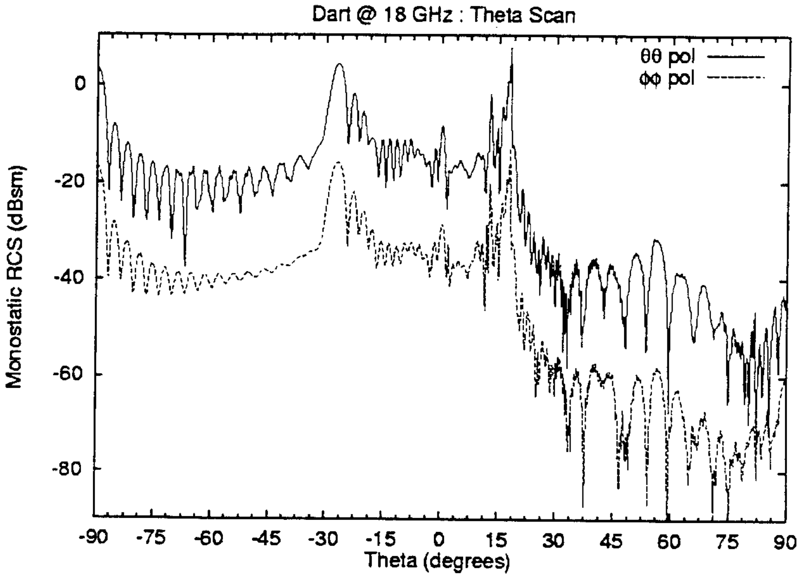


FIG. 3. *RCS of the Dart at 18 GHz in both polarizations using over-the-top scan ($\phi\phi$ polarization shifted -20dB for legibility).*

All of the runs in this section were done in a production environment where FastScat was sharing machine resources with other jobs. Generally the load average did not exceed the number of processors, but this was not always the case.

7 Concluding Remarks

A purpose of this paper is to put forth a more general notion of scalability. Parallel scalability is important since only scalable parallel codes utilize large, expensive computers effectively. But Moore's law and big iron are no match for algorithmic scalability.

The Helmholtz FMM and contemporary large computers are complementary. Consider a slow $O(N^3)$ method with a small prefactor. For these methods, large computers confer little advantage. A modest increase in the number of unknowns quickly exceeds the capacity of even the largest machine. As a result of increased microprocessor performance and microprocessor count (from a few hundred to a few thousand), modern supercomputers are nearly 100 times faster than the Delta. Yet even on these machines, codes that do not take advantage of the algorithmic advances can only do problems about 4 times larger than what the Delta did in 1992. In contrast, the Helmholtz FMM has superior asymptotic complexity but a large prefactor. It takes a fairly big machine just for the FMM to breakeven with respect to the slow method. But the benefit is that you can move out to much larger problems and still stay within the available machine resources. High accuracy solutions for problems exceeding a million square wavelengths are possible on the largest present day machines with modern algorithms.

The FastScat development effort is continuing in the areas of modeling subwavelength structures such as edges and gaps, and in the incorporation of material properties. We see no reason why these extensions can not also be accomplished in a scalable way.

Acknowledgements

We thank Professor Vladimir Rokhlin of Yale University for his guidance over the years. We also thank Dr. Dennis Healy and Dr. Anna Tsao of the Applied and Computational Mathematics Program at DARPA, and Dr. Arje Nachman of the Air Force Office of Scientific Research, for their continued support.

References

[1] L.F. Canino, J.J. Ottusch, M.A. Stalzer, J.L. Visher, and S.M. Wandzura, *Numerical solution of the Helmholtz equation in 2d and 3d using a high-order Nyström discretization*, J. Computational Physics, 146, 1998, pp. 627–663.

[2] R. Coifman, V. Rokhlin, and S. Wandzura, *The fast multipole method: a pedestrian prescription*, IEEE Antennas and Propagation Mag., 3 (35), June 1993, pp. 7–12.

[3] T. Cwik, J. Patterson, and D. Scott, *Electromagnetic scattering calculations on the Intel Touchstone Delta*, Proc. Supercomputing 92, Nov., Minneapolis, MN, pp. 538–542.

[4] M.A. Epton and B. Dembart, *Multipole translation theory for the three-dimensional Laplace and Helmholtz equations*, SIAM J. Scientific Computing, 4 (16), July 1995, pp. 865–897.

[5] L. Greengard and W.D. Grop, *A parallel version of the fast multipole method*, Computers Math. Applic., 20, 1990, pp. 63–71.

[6] M.F. Gyure and M.A. Stalzer, *A prescription for the multilevel Helmholtz FMM*, IEEE Computational Science & Engineering, July-Sept. 1998, pp. 39–47.

[7] V. Rokhlin, *Diagonal form of translation operators for the Helmholtz equation in three dimensions*, Applied and Computational Harmonic Analysis, 1(1), Dec. 1993, pp. 82–93.

[8] J.P. Singh, C. Holt, J.L. Hennessy, and A. Gupta, *A parallel adaptive fast multipole method*, Proc. Supercomputing '93, Nov., Portland, OR, pp. 54–65.

[9] J.M Song and W.C. Chew, *Multilevel fast multipole algorithm for solving combined field equations of electromagnetic scattering*, Microwave and Optical Technology Letters, 10(1), Sept. 1995, pp. 14–19.

[10] M.A. Stalzer, *A parallel fast multipole method for the Helmholtz equation*, Parallel Processing Letters, 5(2), 1995, pp. 263–274.

[11] M.A. Stalzer, *A scalable multilevel Helmholtz FMM for the Origin 2000*, Proc. Ninth SIAM Conf. on Parallel Processing for Scientific Computing, Mar. 1999, San Antonio, TX.

[12] S. Wandzura and H. Xiao, *Quadrature rules on triangles in R^2*, Yale University Research Report YALEU/DCS/RR-1168, Nov. 1998, New Haven, CT.

[13] M.S. Warren and J.K. Salmon, *A parallel, portable and versatile treecode*, Proc. Seventh SIAM Conf. on Parallel Processing for Scientific Computing, Feb. 1995, San Francisco, CA.

[14] N. Yarvin and V. Rokhlin, *A generalized 1D fast multipole method, with applications to filtering of spherical harmonics*, Tech. Report, 1999, Dept. of Computer Science, Yale University, New Haven, CT.

JOURNAL OF COMPUTATIONAL PHYSICS **147**, 594–609 (1998)
ARTICLE NO. CP986104

A Generalized One-Dimensional Fast Multipole Method with Application to Filtering of Spherical Harmonics[1]

Norman Yarvin[2] and Vladimir Rokhlin

Department of Computer Science, Yale University, P.O. Box 208285 Yale Station,
New Haven, Connecticut 06520-8285
E-mail: yarvin@cs.yale.edu, rokhlin@cs.yale.edu

Received June 1, 1998; revised September 22, 1998

The need to filter functions defined on the sphere arises in a number of applications, such as climate modeling, electromagnetic and acoustic scattering, and several other areas. Recently, it has been observed that the problem of uniform resolution filtering on the sphere can be performed efficiently via the fast multipole method (FMM) in one dimension. In this paper, we introduce a generalization of the FMM that leads to an accelerated version of the filtering process. Instead of multipole expansions, the scheme uses special-purpose bases constructed via the singular value decomposition of appropriately chosen submatrices of the filtering matrix. The algorithm is applicable to a fairly wide class of projection operators; its performance is illustrated with several numerical examples. © 1998 Academic Press

Key Words: singular value decompositions; fast algorithms; spherical harmonics.

---

1. INTRODUCTION

The fast multipole method (FMM) [6] is an $O(n)$ algorithm for calculating electrostatic potentials at n points due to a set of n charges. Variants of it exist in one [3, 15], two [6, 9], and three [7] dimensions. While the two- and three-dimensional variants have found direct uses, the one-dimensional version is normally used as a step in the solution of other numerical problems (see, for example, [3]). One such use of the one-dimensional FMM has recently been published by Jakob-Chien and Alpert [10], in an algorithm for the rapid uniform resolution filtering and interpolation of functions on the sphere; that algorithm has uses in the solution of partial differential equations on the sphere [13], in fast algorithms for

---

[1] Supported in part by DARPA/AFOSR under Grant F49620-97-1-0011, and in part by ONR under Grant N00014-96-1-0188.

[2] Corresponding author.

0021-9991/98 $25.00
Copyright © 1998 by Academic Press
All rights of reproduction in any form reserved.

electromagnetic scattering [4], and in several other environments. In this paper, we describe a version of the one-dimensional FMM which has been generalized so as to calculate not only electrostatic potentials, but a wide class of similar kernels, and we describe an accelerated version of the algorithm of [10] in which two subroutine calls to the original one-dimensional FMM are replaced by one call to the generalized FMM.

Formally, this paper describes an algorithm for the following task: given an $n \times m$ matrix P of a certain structure and given a desired accuracy ε, compress P so that its product with a vector can be efficiently computed to that accuracy. The structure the algorithm requires of P is as follows: there must exist numbers $x_1 < x_2 < \cdots < x_m$ and $y_1 < y_2 < \cdots < y_n$ such that, roughly speaking, any submatrix of P which is separated in index space from the line $x_i = y_j$ by a distance greater than its own size has a rank less than some (reasonably small) number r, to the precision ε; the CPU time taken by the algorithm for multiplication of P by a vector is then $O(nr)$. (A rigorous accounting of the execution time of the algorithm is somewhat complicated and is given in Section 3.2.6.) One matrix $P = [p_{ij}]$ which has such a structure is given by the formula

$$p_{ij} = \frac{1}{y_i - x_j} \tag{1}$$

and is the matrix whose multiplication by a vector is implemented by the original one-dimensional versions of the FMM.

This paper is arranged as follows. Section 2 briefly reviews numerical tools used by the algorithm. Section 3 describes the generalized FMM in its basic form. Section 4 describes modifications to the algorithm of Section 3, the principal one of which is the diagonalization of roughly a third of the interaction matrices. Section 5 contains numerical results for the generalized FMM applied to the matrix (1). Section 6 describes modifications to the algorithm of [10] which incorporate the generalized FMM. Finally, Section 7 examines generalizations of the schemes presented in this paper.

2. NUMERICAL PRELIMINARIES

2.1. *Singular Value Decomposition*

The singular value decomposition (SVD) is a ubiquitous tool in numerical analysis, given for the case of real matrices by the following lemma (see, for instance, [14] for more details).

LEMMA 2.1. *For any $n \times m$ real matrix A, there exist an integer p, an $n \times p$ real matrix U with orthonormal columns, an $m \times p$ real matrix V with orthonormal columns, and a $p \times p$ real diagonal matrix $S = [s_{ij}]$ whose diagonal entries are nonnegative, such that $A = USV^*$ and that $s_{ii} \geq s_{i+1,i+1}$ for all $i = 1, \ldots, p-1$.*

The diagonal entries s_{ii} of S are called singular values of A; the columns of the matrix V are called right singular vectors; the columns of the matrix U are called left singular vectors.

2.2. *Least Squares Approximation*

This section contains three lemmas on the least squares approximation of matrices, proven in a more general setting in [15]. In this section and in the remainder of the paper $\mathbb{R}^{n,m}$ will denote the space of all real $n \times m$ matrices, and the matrix norm used will be the Schur or Frobenius norm: that is, for an $n \times m$ real matrix $A = [a_{ij}]$,

$$\|A\| = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij}^2}. \tag{2}$$

LEMMA 2.2. *Suppose A is a $p \times n$ real matrix, B is an $m \times k$ real matrix, and C is a $p \times k$ real matrix, for some m, p, n, and k. Let $A = \tilde{U}_A \tilde{S}_A \tilde{V}_A^*$ be a singular value decomposition of A, and let $B = \tilde{U}_B \tilde{S}_B \tilde{V}_B^*$ be a singular value decomposition of B. Let r be the number of nonzero singular values of A, and let q be the number of nonzero singular values of B. Let U_A and V_A consist of the first r columns of $\tilde{U}_A$ and $\tilde{V}_A$, respectively, and let S_A consist of the first r rows of the first r columns of $\tilde{S}_A$. Let U_B and V_B consist of the first q columns of $\tilde{U}_B$ and $\tilde{V}_B$, respectively, and let S_B consist of the first q rows of the first q columns of $\tilde{S}_B$. Then the solution $\hat{X}$ of the minimization problem,*

$$\min_{X \in \Xi^{n,m}} \|AXB - C\|, \tag{3}$$

is given by

$$\hat{X} = V_A S_A^{-1} U_A^* C V_B S_B^{-1} U_B^*. \tag{4}$$

Furthermore,

$$\|A\hat{X}B - C\| = \|C - U_A U_A^* C V_B V_B^*\|. \tag{5}$$

The following lemma provides a bound, in certain situations, on the error of the approximation given by Lemma 2.2.

LEMMA 2.3. *Under the conditions of Lemma 2.2, suppose that there exist an $n \times k$ matrix D and an $p \times m$ matrix E such that*

$$\|AD - C\| < \varepsilon_1 \tag{6}$$

and

$$\|EB - C\| < \varepsilon_2. \tag{7}$$

Then

$$\|A\hat{X}B - C\| < \varepsilon_1 + \varepsilon_2. \tag{8}$$

As shown by the following lemma, the error bound of Lemma 2.3 also applies when a different formula for the minimizing matrix is used.

LEMMA 2.4. *Under the conditions of Lemma 2.3, let the $n \times m$ matrix Y be given by the formula*

$$Y = D V_B S_B^{-1} U_B^*. \tag{9}$$

Then

$$\|AYB - C\| < \varepsilon_1 + \varepsilon_2. \tag{10}$$

3. BASIC FMM

This section describes the generalized FMM of this paper. It is described as a set of modifications to the FMM of [6, 3]; the reader is assumed to be familiar with that algorithm.

The overall FMM structure of an upward pass for creation of far field expansions, followed by a pass which computes local expansions from far field expansions, followed by a downward pass which propagates local expansions to lower levels and evaluates them, is retained. However, all the expansions are different, being based on singular value decompositions rather than on analytical formulae. In addition, the hierarchical subdivision scheme is different, being performed according to matrix indices rather than according to point locations. (The expansions used permit almost any subdivision scheme, whether adaptive as in [15], or nonadaptive as in [3]; the present scheme was chosen solely for its simplicity.)

3.1. *Subdivision Scheme*

The hierarchical subdivision is performed on column indices of the matrix P, as follows:

- Each interval of column indices, if it is divided, is divided into two intervals of equal size (or differing in size by one, if the number of indices in the interval is odd).
- The subdivision is uniform; either all the intervals at any given depth of the tree are subdivided, or none are.
- The subdivision process continues until the lowest-level intervals are as close as possible to a user-chosen size.

For each interval $[j_1, j_2]$ of column indices produced by the above process, a corresponding interval $[i_1, i_2]$ of row indices is chosen such that the portion of P addressed by the two intervals of indices contains as much as possible of the line $x_i = y_j$. The precise criterion used to choose the interval $[i_1, i_2]$ is that it should be the interval of maximal size such that

$$\left(x_{j_1-1} + x_{j_1}\right)/2 \le y_{i_1} < \cdots < y_{i_2} < \left(x_{j_2} + x_{j_2+1}\right)/2. \tag{11}$$

(If x_{j_1-1} or x_{j_2+1} does not exist, the corresponding inequality in the above equation is not enforced. The quantities $x_1 < x_2 < \cdots < x_m$ and $y_1 < y_2 < \cdots < y_n$ were, in the present implementation, user-provided; in an environment where they are not readily available, they can be determined by numerically searching P for areas of high numerical rank.)

3.2. *Expansions*

This section describes the expansions used in the generalized FMM. Submatrices of P will be designated as follows: $P_{a,b}$ denotes the portion of P whose column indices are in b and whose row indices are in a, where a and b are either intervals of indices into P, or sets thereof.

For each interval, the FMM divides the intervals at the same depth in the tree into two sets:

- 1. The *near field* region, consisting of the interval itself and the two adjacent intervals at the same depth in the tree of intervals.
- 2. The *far field* region, consisting of all remaining intervals at the same depth in the tree. We denote the far field region of the i'th interval by F_i.

A third set is also required: the *interaction list* of an interval i is the set of intervals at the same depth in the tree which are in the far field of i and which are not in the far field of the parent of i.

3.2.1. Far-field expansions. The original FMM [6] relies on the fact that the electrostatic potential due to a set of charges can be represented to high precision, at points distant from those charges, by a multipole expansion of relatively few terms. In the generalized FMM described in this paper, the output (no longer necessarily the electrostatic potential, although we will continue to use the terms "potential" and "charge" for convenience) does not need to be describable by a multipole expansion, but can be describable by an arbitrary expansion, provided that the expansion coefficients are linear functions of the charge magnitudes and that the potential is a linear function of the expansion coefficients. The creation and evaluation matrices for this expansion, which we will call a far-field expansion, do not need to be furnished as such by the user; they are computed from the matrix P using the singular value decomposition. This computation is performed for each interval i for which a far-field expansion is needed and is as follows: Let $n_i \times m_i$ be the dimensions of the matrix $P_{F_i,i}$, let the singular value decomposition of $P_{F_i,i}$ be denoted by $\tilde{U}\tilde{S}\tilde{V}^*$, the number of singular values by $\tilde{p}$, and the singular values by $s_1 \geq s_2 \geq \cdots \geq s_{\tilde{p}}$. Let p_i be the minimum integer such that

$$\sum_{j=p_i+1}^{\tilde{p}} s_j^2 < \varepsilon^2 \|P\|^2 \frac{n_i m_i}{nm}. \tag{12}$$

Let the $m_i \times p_i$ matrix V_i consist of the first p_i columns of $\tilde{V}$ and let the $p_i \times n_i$ matrix E_i consist of the first p_i columns of the product $\tilde{U}\tilde{S}$. We will refer to V_i^* as the far-field expansion creation matrix for interval i and to E_i as the far-field evaluation matrix; the latter is not used explicitly in the algorithm.

As shown in [8], the product $E_i V_i^*$ is, among matrices of rank p_i, the closest approximation to the matrix $P_{F_i,i}$ in the norm (2). Thus the number of terms in any known expansion for $P_{F_i,i}$ (such as a multipole expansion) is an upper bound for the number of terms p_i in the far-field expansion of the same accuracy computed as above.

3.2.2. Local expansions. Using far-field expansions alone, an $O(n \cdot \log n)$ version of the FMM can be produced (for an overview of the various versions see [7]). The $O(n)$ version of the FMM requires additional numerical machinery, namely local expansions, which approximate the potential on a region due to charges on distant regions. In the original FMM, local expansions were harmonic expansions; in the generalized FMM, creation and evaluation matrices for local expansions are computed from the matrix P using the singular value decomposition, as follows. Let $n_i' \times m_i'$ be the dimensions of the matrix P_{i,F_i}; let the singular value decomposition of P_{i,F_i} be denoted by $\tilde{U}\tilde{S}\tilde{V}^*$, the number of singular values by $\tilde{r}$, and the singular values by $s_1 \geq s_2 \geq \cdots \geq s_{\tilde{r}}$. Let r_i be the minimum integer such that

$$\sum_{j=r_i+1}^{\tilde{r}} s_j^2 < \varepsilon^2 \|P\|^2 \frac{n_i' m_i'}{nm}. \tag{13}$$

Let the $m_i' \times r_i$ matrix U_i consist of the first r_i columns of $\tilde{U}$. We will refer to U_i as the local expansion evaluation matrix for interval i.

3.2.3. Far-field translation matrices. The FMM does not compute far-field expansions for intervals at high levels in the tree directly from the charges in the interval, but rather computes them from far-field expansions at lower levels. Associated with each interval i whose parent interval j has a far-field expansion is a translation matrix T_i which takes as input a far-field expansion for i and produces as output a far-field expansion for j which evaluates to the same potential. Let V_i^* be the far-field creation matrix for interval i, and let $V_{j,i}^*$ be the far field creation matrix for interval j, with columns deleted such that it only accepts input from the interval i. Clearly the translation matrix T_i should be such that for any m_i-vector q, the vector $T_i V_i^* q$ is as close as possible, by some measure, to the vector $V_{j,i}^* q$. The measure we use is the least squares measure: in particular, T_i is chosen so as to minimize the quantity $\|V_{j,i}^* - T_i V_i^*\|$. The formula for such minimization is given by Lemma 2.2: using the fact that the singular value decomposition of any matrix with orthogonal columns consists of that matrix multiplied by two identity matrices, it reduces in this case to

$$T_i = V_{j,i}^* V_i. \tag{14}$$

We will refer to T_i as the far-field expansion translation matrix for interval i.

Lemma 2.4 gives a bound for the error associated with using the translation matrix T_i. Suppose $E_{j,k}$ and $E_{i,k}$ are matrices which take as input the far-field expansions on interval j and on interval i, respectively, and use them to evaluate the potential on some other interval k and are such that

$$\|P_{i,k} - E_{j,k} V_{j,i}^*\| < \varepsilon_1 \tag{15}$$

$$\|P_{i,k} - E_{i,k} V_i^*\| < \varepsilon_2. \tag{16}$$

Using (15), (16), and Lemma 2.4, we get that

$$\|P_{i,k} - E_{j,k} T_i V_i^*\| < \varepsilon_1 + \varepsilon_2. \tag{17}$$

3.2.4. Local expansion translation matrices. The FMM does not evaluate local expansion for intervals at high levels in the tree directly at each of the points at which the potential is to be evaluated, but rather transforms them into local expansions for intervals at lower levels. Associated with each interval i, whose parent interval j has a local expansion, is a translation matrix M_i which takes as input a local expansion on j and produces as output a local expansion on i. M_i is computed as follows. Let U_i be the local expansion evaluation matrix for interval i, and let $U_{j,i}$ be the local expansion evaluation matrix for interval j, with rows deleted so that it only produces output on the interval i. Clearly the translation matrix M_i should be such that for any r_i-vector α, the vector $U_i M_i \alpha$ is as close as possible, by some measure, to the vector $U_{j,i}\alpha$. The measure we use is the least squares measure; in particular, M_i is chosen so as to minimize the quantity $\|U_{j,i} - U_i M_i\|$. The formula for such minimization is given by Lemma 2.2. Using the fact that the singular value decomposition of any matrix with orthogonal columns consists of that matrix multiplied by two identity matrices, it reduces in this case to

$$M_i = U_i^* U_{j,i}. \tag{18}$$

The error incurred by using M_i is bounded by Lemma 2.4; the analysis is almost identical to that presented in Section 3.2.3 for the far-field translation matrix T_i and is omitted. We will refer to M_i as the local expansion translation matrix for interval i.

3.2.5. Far-field to local interaction matrices. A far-field to local interaction matrix $E_{j,i}$ takes as input a far-field expansion on an interval i and produces as output a local expansion on another interval j. Such matrices are constructed only for pairs of intervals (i, j) such that j is in the interaction list of i. The matrix $E_{j,i}$ should be such that for all m_i-vectors q the product $U_j E_{j,i} V_i^* q$ is as close as possible, by some measure, to the product $P_{j,i} q$. We choose $E_{j,i}$ so as to minimize the quantity

$$\varepsilon_{j,i} = \| U_j E_{j,i} V_i^* - P_{j,i} \|. \tag{19}$$

The formula for such minimization is given by Lemma 2.2; using the fact that the singular value decomposition of any matrix with orthogonal columns consists of that matrix multiplied by two identity matrices, it reduces in this case to

$$E_{j,i} = U_j^* P_{j,i} V_i. \tag{20}$$

Lemma 2.3, combined with (12) and (13), gives a bound for $\varepsilon_{j,i}$:

$$\varepsilon_{j,i} < \varepsilon \| P \| \left(\sqrt{\frac{n_i m_i}{nm}} + \sqrt{\frac{n_i' m_i'}{nm}} \right). \tag{21}$$

We will refer to $E_{j,i}$ as the far field to local interaction matrix from interval i to interval j.

Remark 3.1. A brief inspection of the above formulae for the creation, translation, and evaluation matrices $\{U_i\}$, $\{V_i\}$, $\{T_i\}$, $\{M_i\}$, and $\{E_{j,i}\}$ shows that the same matrices are generated, in different roles, if the input matrix to the algorithm is the adjoint P^* of P, provided that the hierarchical subdivision is retained: the far field expansion creation matrices for P are identical to the local expansion evaluation matrices for P^*, and vice versa; the far field translation matrices for P are identical to the local expansion translation matrices for P^*, and vice versa; and the far field to local matrices for P are the adjoints of the far field to local matrices for P^*. Thus the matrices precomputed for P can also be used for multiplying by P^*.

3.2.6. Execution time. The FMM performs one matrix–vector multiplication for each instance of the matrices $\{U_i\}$, $\{V_i\}$, $\{T_i\}$, $\{M_i\}$, and $\{E_{j,i}\}$. Thus the CPU time which it consumes is proportional to the total number of elements in all instances of the matrices. The sizes of the matrices depend on the numerical ranks p_i and r_i, as defined by (12) and (13). We analyze the execution time further only in the case that all those ranks are all bounded by some number r. In that case, the computation of far-field expansions from the input takes $O(mr)$ time, the computation of the output from local expansions takes $O(nr)$ time, and the computations of expansions from other expansions take $O(kr^2)$ time, where k is the total number of intervals produced by the subdivision process. Assuming that m is proportional to n, the total execution time is $O(nr + kr^2)$. The quantity $nr + kr^2$ is minimized (with respect to k) when n/k is equal to r. Since n/k is proportional to the size of the lowest-level intervals, the minimum execution time occurs when the size of the lowest-level intervals is proportional to r, with the constant of proportion depending on the details of the computer involved.

4. TECHNICAL IMPROVEMENTS

4.1. *Diagonalization of Far Field to Local Matrices*

A certain amount of freedom is present in the definition of far field and local expansions: the results of the FMM are clearly unaffected if the far-field expansion creation matrix V_i^* for an interval i is multiplied on the left by any orthogonal matrix W, its far field translation matrix T_i is multiplied on the right by W^*, and its far field to local matrices $E_{j,i}$ for all j are multiplied on the right by W^*. Similarly, the results of the FMM are unaffected if the local expansion evaluation matrix U_i for an interval i is multiplied on the right by any orthogonal matrix W, its local expansion translation matrix M_i is multiplied on the left by W^*, and its far field to local matrices $E_{i,j}$ for all j are multiplied on the left by W^*.

We use this freedom to diagonalize one of the (usually three) far field to local matrices for each interval. Suppose that $E_{i,j}$ for some intervals i and j is the matrix to be diagonalized. Let its singular value decomposition be denoted by $E_{i,j} = USV^*$. Then we multiply V_j^* on the right by V^*, and multiply U_i on the left by U, also changing translation matrices and far field to local matrices as indicated in the previous paragraph so that the results of the FMM are unaffected.

Far field to local matrices are chosen for diagonalization in such a way that each expansion redefined by this process is redefined only once. The scheme used is as follows: each level of intervals is divided into blocks of four adjacent intervals; inside each block the interactions chosen for diagonalization are: 1 → 3, 2 → 4, 3 → 1, and 4 → 2 (as depicted in Fig. 1).

4.2. *Splits by Factors Other Than Two*

Another modification which was made to the above FMM is to split intervals into more than two pieces. This clearly can be done to any interval, at any level in the tree. However, the only use which was made of this flexibility was to alter the top of the tree of intervals slightly, so as to control better the size of the lowest-level intervals in the tree. The top interval was split either into two, three, or five pieces; if three, its subintervals might each

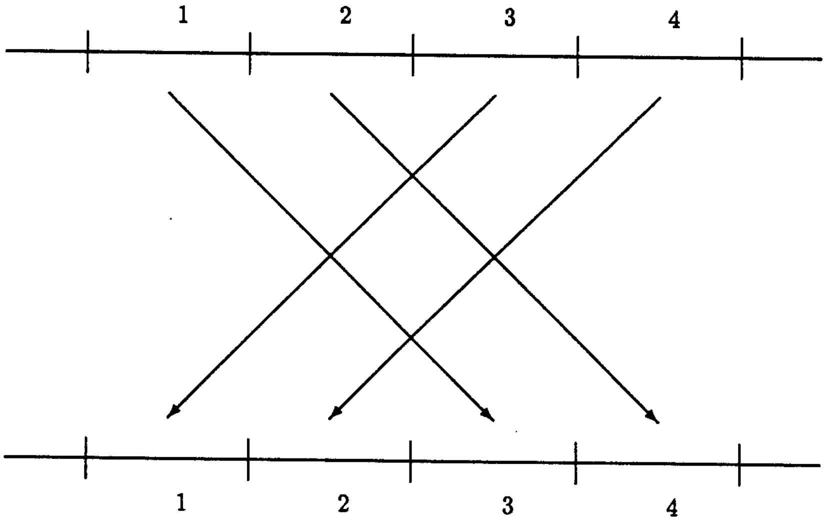


FIG. 1. Far field to local operators which are diagonalized.

TABLE I
Double Precision Timings for the 1/*x* Kernel

N	Error (L^2 norm)	Times (seconds) Init	Eval	Direct	Ratio eval/FFT	Memory (REAL*8 spaces)
64	0.35477E-15	0.070	0.001	0.001	5.21	3852
128	0.92042E-15	0.820	0.003	0.005	7.31	10407
256	0.23512E-14	6.620	0.007	0.019	8.93	26205
512	0.16144E-13	39.700	0.013	0.073	5.60	52263
1024	0.21925E-13	214.710	0.031	0.730	4.16	117881

be split into three parts, the remaining intervals in the tree all being split into two parts. This permits a choice of the size of the lowest-level intervals not only of $n/2^k$ for any k, but also of $n/(3 \times 2^k)$, $n/(5 \times 2^k)$, or $n/(9 \times 2^k)$.

5. NUMERICAL RESULTS

For comparison against the older one-dimensional FMMs of [3, 15], the generalized FMM was applied to the $1/x$ kernel: that is, the input matrix $P = [p_{ij}]$ was given by (1). Timings for various numbers of points n are listed in Tables I and II for double and single precision (that is, with the parameter ε set to 10^{-14} and 10^{-7}). In all cases, the parameter m was set to be equal to n, the nodes $\{x_i\}$ were identical to the nodes $\{y_i\}$, being slightly perturbed equispaced nodes. All timings were performed on a Sun Sparcstation 10 in double precision (Fortran REAL*8) arithmetic. Also included in the tables are ratios of the execution time of the algorithm to the execution time of a standard SLATEC FFT of size n.

From the timings, it can be seen that the generalized FMM is similar in execution speed to the best previous 1D FMM (that of [15]) known to the authors. It is, however, far inferior to the FMMs of [3, 15] in the time spent in the precomputation stage; initialization times for those algorithms did not exceed execution time by more than a factor of 10, whereas the initialization time for the generalized FMM exceeds the execution time by factors of 1000s. Effectively, it limits the usefulness of the procedure of this paper to problems of sufficient importance that the initialization data can be precomputed and stored. The following section discusses one such case.

TABLE II
Single Precision Timings for the 1/*x* Kernel

N	Error (L^2 norm)	Times (seconds) Init	Eval	Direct	Ratio eval/FFT	Memory (REAL*8 spaces)
64	0.25040E-08	0.040	0.001	0.001	4.74	3500
128	0.23352E-07	0.440	0.002	0.005	5.90	8465
256	0.19125E-06	3.580	0.005	0.018	6.13	17803
512	0.64886E-06	22.710	0.010	0.074	4.03	36911
1024	0.28910E-06	124.690	0.021	0.590	2.77	79407

6. APPLICATION TO FILTERING

This section describes a use of the generalized FMM, in an algorithm recently published by Jakob-Chien and Alpert [10] for uniform resolution filtering of functions on the sphere. Their algorithm as a whole performs the following task: given numbers $f(\phi_i, \theta_j)$, $i = 1, \dots, I$; $j = 1, \dots, J$, such that

$$f(\phi_i, \theta_j) = \sum_{n=0}^{K} \sum_{m=-n}^{n} f_n^m Y_n^m(\phi_i, \theta_j), \tag{22}$$

computes numbers $\tilde{f}(\tilde{\phi}_i, \tilde{\theta}_j)$ such that

$$\tilde{f}(\tilde{\phi}_i, \tilde{\theta}_j) = \sum_{n=0}^{N} \sum_{m=-n}^{n} f_n^m Y_n^m(\tilde{\phi}_i, \tilde{\theta}_j), \tag{23}$$

where the functions Y_n^m are the surface harmonics and where $\{\phi_i\}$, $\{\theta_j\}$, $\{\tilde{\phi}_i\}$, and $\{\tilde{\theta}_j\}$ are appropriately chosen grid points (see [10] for details).

We modify only the core of the algorithm of [10], which performs the following one-dimensional filtering operation: given numbers $f^m(\theta_1), \dots, f^m(\theta_J)$ such that

$$f^m(\theta_i) = \sum_{j=m}^{J-1} f_j^m \bar{P}_j^m(\mu_i), \quad i = 1, \dots, J, \tag{24}$$

compute numbers $\tilde{f}^m(\tilde{\theta}_1), \dots, \tilde{f}^m(\tilde{\theta}_N)$ such that

$$\tilde{f}^m(\tilde{\theta}_i) = \sum_{j=m}^{N} f_j^m \bar{P}_j^m(\tilde{\mu}_i), \quad i = 1, \dots, N, \tag{25}$$

where the functions $\bar{P}_n^m$ are the normalized associated Legendre functions, $\mu_i = \sin\theta_i$ and $\tilde{\mu}_i = \sin\tilde{\theta}_i$.

Due to the orthonormality of the functions $\bar{P}_n^m$ for fixed m and integer $n \geq m$, if the nodes $\mu_1, \dots, \mu_J$ are Legendre nodes (nodes of the Gaussian quadrature corresponding to the weight function $\omega(x) = 1$; see, for instance, [14]), then the coefficients $f_m^m, f_{m+1}^m, \dots, f_N^m$ are given by

$$f_n^m = \sum_{j=1}^{J} f^m(\theta_j) \bar{P}_n^m(\mu_j) w_j, \tag{26}$$

where $w_1, \dots, w_J \in \mathbb{R}$ are the Gaussian weights corresponding to the nodes $\mu_1, \dots, \mu_J$. Combining (25) and (26) yields an equation for the entire filtering operation:

$$\tilde{f}^m(\tilde{\theta}_i) = \sum_{k=1}^{J} f^m(\theta_k) w_k \sum_{j=m}^{N} \bar{P}_j^m(\mu_k) \bar{P}_j^m(\tilde{\mu}_i). \tag{27}$$

Equation (27) constitutes a linear transformation from $f^m(\theta_1), \dots, f^m(\theta_J)$ to $\tilde{f}^m(\tilde{\theta}_1), \dots, \tilde{f}^m(\tilde{\theta}_N)$; we will refer to the matrix of this transformation as the filtering matrix and will denote it by P. Using the Christoffel–Darboux formula for the associated Legendre functions (see, for instance, [1, Section 8.9.1]), which is

$$(\tilde{\mu}-\mu)\sum_{n=|m|}^{N}\bar{P}_n^m(\tilde{\mu})\bar{P}_n^m(\mu)=\varepsilon_{N+1}^m\left(\bar{P}_{N+1}^m(\tilde{\mu})\bar{P}_N^m(\mu)-\bar{P}_N^m(\tilde{\mu})\bar{P}_{N+1}^m(\mu)\right). \tag{28}$$

where

$$\varepsilon_n^m=\sqrt{(n^2-m^2)/(4n^2-1)}, \tag{29}$$

the filtering operation can be written as

$$\frac{\tilde{f}^m(\tilde{\theta}_j)}{\varepsilon_{N+1}^m}=\bar{P}_{N+1}^m(\tilde{\mu}_j)\sum_{i=1}^{J}\frac{f^m(\theta_i)w_i\bar{P}_N^m(\mu_i)}{\tilde{\mu}_j-\mu_i}-\bar{P}_N^m(\tilde{\mu}_j)\sum_{i=1}^{J}\frac{f^m(\theta_i)w_i\bar{P}_{N+1}^m(\mu_i)}{\tilde{\mu}_j-\mu_i}. \tag{30}$$

From (30) it immediately can be seen that the filtering matrix consists of the sum of two matrices of the form (1), each multiplied on the left and the right by a diagonal matrix. Thus, the filter can be implemented using two calls to an FMM for the $1/x$ kernel; this is the method presented in [10] (from where the above analysis is copied). It also follows that, if the generalized FMM of this paper is applied to the filtering matrix, the numerical ranks $\{r_i\}$ and $\{p_i\}$ (see (13) and (12)) are no more than twice the corresponding ranks when the generalized FMM is applied to a matrix of the form (1). Thus, the filter can be implemented efficiently via a single call to the generalized FMM.

Remark 6.1. If N is larger than J, the operation (30) amounts to interpolation rather than filtering. If the output nodes $\{\tilde{\mu}_i\}$ are the Legendre nodes of order N, then the filtering matrix from J nodes to N nodes is, except for the multiplication of the input by Gaussian weights, the adjoint of the interpolation matrix from N nodes to J nodes; this can easily be seen by inspection of (30). Thus, the matrices $\{U_i\}$, $\{V_i\}$, $\{T_i\}$, $\{M_i\}$, and $\{E_{j,i}\}$, precomputed for the purpose of filtering, can also be used for interpolation (see Remark 3.1).

6.1. *General Nodes*

If the nodes $\mu_1,\ldots,\mu_J$ are not Legendre nodes, then the coefficients $f_m^m,\ldots,f_N^m$ cannot be computed by direct use of the formula (26). In this case, two methods of performing the filtering operation are available. First, Eq. (24) can be solved for the coefficients $f_m^m,\ldots,f_J^m$. Alternatively, the function can be interpolated onto Legendre nodes, following which the filtering matrix for Legendre nodes (30) can be used. We use the second method to show that the filtering matrix for general nodes can be compressed by the generalized FMM; we used the first method in our implementation.

As is well known (see, for instance, [1]), each of the associated Legendre functions P_n^m is either a polynomial or a polynomial multiplied by $\sqrt{1-x^2}$, depending on whether m is even or odd. Thus the interpolation onto Legendre nodes is a polynomial interpolation, which, if m is odd, is preceded by a division by $\sqrt{1-x^2}$ and followed by a multiplication by $\sqrt{1-x^2}$. As shown in [3], polynomial interpolation can be performed in $O(n)$ time using an FMM. The filtering matrix for general nodes is the product of the interpolation matrix and the filtering matrix for Legendre nodes; since each of these can be compressed by a generalized FMM, their product also can be compressed by a generalized FMM (see [2]).

Remark 6.2. In the solution of Eq. (24) for the coefficients $f_m^m, \ldots, f_N^m$, when $m > 0$, there are more equations than unknowns. The definition of the problem is such that there is an exact solution; however, numerically, this issue was dealt with by solving the equation in the least squares sense.

6.2. *Optimizations*

The above filtering algorithm admits several optimizations. We describe them only for the case when the nodes $\mu_1, \ldots, \mu_J$ are Legendre nodes; however, all of them have also been implemented in the case of general nodes.

First, when m is close to N, the number of coefficients f_j^m to be extracted is small; thus direct computation of (26) followed by (25) is the most efficient algorithm for the filter.

Second, portions of the filtering matrix have negligible norm and can be discarded. This can be easily seen by examination of (30), using the fact that the functions P_n^m take on small values near the endpoints of the interval $[-1, 1]$. The fraction of the matrix which can be discarded increases with increasing m, to as much as eight ninths. This optimization is clearly not specific to the generalized FMM; it can be applied equally well to the direct method or to the unaltered algorithm of [10] and was applied to the direct method code which was used in the timings presented below.

Third, the filter can be speeded up slightly by splitting the input function into odd and even parts, and filtering them separately. Each of the associated Legendre functions P_n^m is either odd or even, with functions of successive degree n being alternately odd and then even. Thus the filter, applied to an odd function, yields an odd function and, applied to an even function, yields an even function. This implies that the filtering matrix is block-diagonalized (into two blocks) by the separation of odd functions from even functions. We address only the case in which the separation can be done trivially, that is, when each of the sets of nodes $\{\mu_i\}$ and $\{\tilde{\mu}_i\}$ is symmetric around zero; for brevity of explanation, we further assume that N and J are even. In this case the separation of odd functions from even functions is accomplished by the usual formulae

$$f_{\text{odd}}(x) = (f(x) - f(-x))/2, \tag{31}$$

$$f_{\text{even}}(x) = (f(x) + f(-x))/2, \tag{32}$$

where, as usual, each of the functions f_{odd} and f_{even} are symmetric around zero and, thus, need only be stored at half the nodes. It is easily shown, using (30) and (31), that in the case that the nodes $\mu_1, \ldots, \mu_J$ are Legendre nodes, each block $\hat{P} = [\hat{p}_{ij}]$ of the block-diagonalized filtering matrix is given by

$$\begin{aligned}\hat{p}_{ij} = {} & \frac{\bar{P}_{N+1}^m(\tilde{\mu}_j)\bar{P}_N^m(\mu_i)w_i - \bar{P}_N^m(\tilde{\mu}_j)\bar{P}_{N+1}^m(\mu_i)w_i}{\tilde{\mu}_j - \mu_i} \\ & \pm \frac{\bar{P}_{N+1}^m(\tilde{\mu}_j)\bar{P}_N^m(\mu_i)w_i + \bar{P}_N^m(\tilde{\mu}_j)\bar{P}_{N+1}^m(\mu_i)w_i}{\tilde{\mu}_j + \mu_i},\end{aligned} \tag{33}$$

where, for the block which filters even functions, the "±" sign is an addition, and, for the block which filters odd functions, it is a subtraction. An inspection of (33) immediately shows that each block is compressible by a generalized FMM.

Remark 6.3. Experimentally, the ranks produced by the generalized FMM when applied to the block-diagonalized matrix are almost identical to the ranks produced when applied to the original filtering matrix, except near the point $\mu = 0$, where the ranks are slightly smaller in the block-diagonalized version.

Remark 6.4. Since the generalized FMM is, when applied to matrices of this form, an $O(n)$ procedure, splitting the problem into two problems of half the size does not produce any asymptotic improvement in execution time, although it does produce an improvement for small to medium-sized n. By contrast, applying this optimization to the direct method (as was done in the code used in the timings presented below) reduces the execution time by a factor of 2 asymptotically, since the direct method is $O(n^2)$.

6.3. *Numerical Results*

Table III contains experimental results for the filter for functions tabulated at Legendre nodes. The filter was run for several values of J, with $N = J/2$ and for each $m = 1, \ldots, N$: the average initialization and execution times, the average L^2 error, and the average amount of memory used for precomputed data (for all values of m) are tabulated. The quantity labeled as initialization time is, as before, the amount of time taken to compute the matrices which comprise the generalized FMM; this task only needs to be performed once for any combination of J and N, since the precomputed matrices can be stored. All figures were produced by an implementation in double precision (Fortran `REAL*8`) arithmetic on a Sun Sparcstation 10. The table also contains the amount of time taken by the direct method and the ratio of the execution time of the FMM-based filter to the execution time of a standard

TABLE III
Filter Timings for Points Tabulated at Legendre Nodes

	Average time per m (seconds) for					
J	Direct	FMM eval	FMM init	Ratio: eval/FFT	Average error (L^2)	Average memory used (REAL*8 spaces)
			Requested accuracy 10^{-3}			
64	0.00014	0.00021	0.038	1.10	0.87216E-04	637
128	0.00059	0.00063	0.173	1.73	0.21141E-03	1814
256	0.00239	0.00172	0.861	2.25	0.35270E-03	4684
512	0.00916	0.00406	4.528	1.64	0.55393E-03	10586
1024	0.15601	0.00930	22.708	1.26	0.72021E-03	22799
			Requested accuracy 10^{-7}			
64	0.00016	0.00020	0.035	1.05	0.62995E-09	715
128	0.00069	0.00068	0.145	1.84	0.89805E-08	2351
256	0.00272	0.00199	0.749	2.61	0.20946E-07	7074
512	0.01015	0.00545	4.480	2.21	0.35158E-07	18763
1024	0.17623	0.01351	25.102	1.84	0.50011E-07	45001
			Requested accuracy 10^{-12}			
64	0.00017	0.00018	0.035	0.97	0.64733E-13	712
128	0.00078	0.00070	0.118	1.88	0.36187E-12	2604
256	0.00312	0.00221	0.630	2.90	0.13528E-12	8496
512	0.01102	0.00656	3.752	2.64	0.30608E-12	26072
1024	0.19227	0.01763	26.347	2.37	0.14238E-11	66714

SLATEC FFT of size J. The direct method for which timings are listed is a modestly optimized variant: the filtering matrix it used was precomputed: certain optimizations used for the FMM-based method were also applied to it, as described in Section 6.2.

The filter was also implemented for functions tabulated at general nodes (Section 6.1) and was tested on Chebyshev nodes. The timings are almost identical, with the only major difference being that considerably more time was required to compute the filtering matrix: they are omitted.

Remark 6.5. The implausibly large CPU times taken by the direct method for $J = 1024$ are the result of the problem size exceeding the size of the cache: on the machine on which timings were run, only two double precision vectors of length 1024 fit in the data cache. Such a jump in timings is not expected to occur on most machines and, in any case, could be eliminated by use of a blocked matrix–vector multiplication routine.

Figure 2 is a graph of the average numerical rank of interaction found by the filter for Legendre nodes (the average of the ranks $\{p_i\}$), plotted as a function of m, for $J = 1024$ and $\varepsilon = 10^{-12}$. (The ranks for the filter for arbitrary nodes, when applied to Chebyshev nodes,

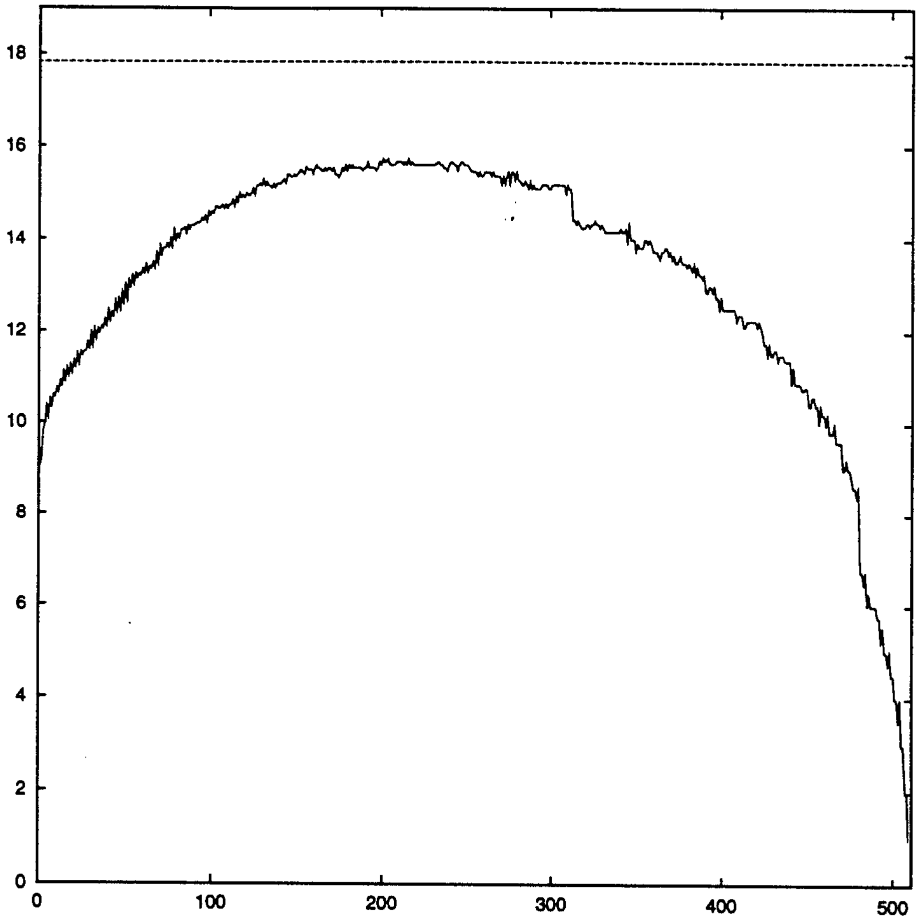


FIG. 2. Average numerical rank of interaction, as a function of m, for $J = 1024$ and $\varepsilon = 10^{-12}$. The dashed line is the theoretical bound on the rank.

were nearly identical.) Also plotted in Fig. 2 is the theoretical upper bound for the average rank. that is. twice the average rank of an FMM for the $1/x$ kernel of the same accuracy. Since most of the ranks were close to their average. the execution time of the FMM is roughly proportional to the average rank. (See Section 3.2.6 for an analysis of the case of all ranks being equal: a similar analysis applies to other variants of the 1D FMM.) Thus. Fig. 2 provides a rough indication of the amount of speedup that is obtained by switching from the scheme of [10] to the generalized FMM: to a first approximation. if the average rank were equal to its upper bound for all m. the two schemes would be of equal speed: to the extent that it is lower. the generalized FMM is faster. (However. it should be noted that the generalized FMM requires more precomputed data and is. thus. more vulnerable to caching effects.)

7. GENERALIZATIONS

In this paper. we have presented a scheme for the efficient filtering of functions on the two-dimensional sphere. The approach is based on two observations. The first observation is that in the fast multipole method (see. for example. [3. 6]) potential kernels can be replaced with functions from a much more general class. using the standard singular value decomposition. and that this yields a fairly efficient implementation. The second observation is that the Christoffel–Darboux formula (28) provides a straightforward proof that the filtering operator on the sphere (27) can be compressed by FMM-type techniques. Both observations admit far-reaching generalizations. outlined below.

1. The fast multipole method used in this paper is a special case of an extremely general procedure. Particular versions of this procedure have been used repeatedly (see [11. 12]): it is effective in all situations when the operator can be compressed by wavelet techniques. The following is a brief outline of the approach.

Given a matrix to be rapidly applied to arbitrary vectors. examine it (either analytically or numerically). identifying large submatrices that are of low rank. When the coefficients of a submatrix are a sufficiently smooth function of its indices. such a submatrix is guaranteed to have a low rank (this is the environment where wavelets and wavelet-type techniques can be used): another frequently encountered situation involves submatrices that are not smooth. but are smooth matrices multiplied by diagonal matrices from the left and/or from the right (as in the case of the filtering operator (30)). Any matrix whose rank is much lower than its dimensionality is "compressed" by its singular value decomposition: applying this procedure to a sufficiently large collection of submatrices of some matrix. we obtain a primitive "fast" algorithm for applying it to arbitrary vectors. The scheme is further accelerated by recursive application of this approach.

A strong argument can be made that the SVD of a matrix is its "optimal" low-rank representation: in this sense. SVD-based implementations of FMM-type algorithms are "optimal." Indeed. schemes have been constructed using the SVD to further compress multipole expansions (see. for example. [3. 9]): the resulting procedures tend to be more efficient than the original FMM. In addition. the FMM for potential kernels has been accelerated (dramatically so. in higher dimensions) by using diagonal forms of translation operators (see [7. 15]). Possible hybrid algorithms combining the latter with SVD-based compression of more general kernels are currently under investigation in one. two. and three dimensions.

2. Formula (28) in the present paper is a special case of the well-known Christoffel–Darboux formula,

$$\sum_{k=0}^{n} p_k(x) \cdot p_k(y) = \frac{q_n}{q_{n+1}} \cdot \frac{p_{n+1}(x) \cdot p_n(y) - p_{n+1}(y) \cdot p_n(x)}{x - y}. \tag{34}$$

where p_k are polynomials orthogonal with *some* weight function w on *some* interval, q_k is the coefficient at the term x^k in the polynomial p_k, and n is an arbitrary positive integer (see, for example, [5, Section 8.902]). It is immediately clear from (34) that the algorithm of this paper can be used to evaluate rapidly the projections in spaces of polynomials on subspaces consisting of polynomials of reduced rank, in the norm associated with the weight w. There are a number of other projections that can be evaluated rapidly using the FMM scheme of this paper, or its variants. The operators we have experimented with include projections on subspaces in the space of polynomials in two dimensions, projections on subspaces spanned by appropriately chosen Bessel functions, and several others. In some cases, we have determined experimentally that the scheme works, but have not constructed the underlying mathematics. This whole class of issues is currently under investigation.

REFERENCES

1. M. Abramowitz and I. Stegun, *Handbook of Mathematical Functions*, Appl. Math. Ser. (Natl Bureau of Standards, Washington, DC, 1964).
2. B. Alpert, G. Beylkin, R. Coifman, and V. Rokhlin, Wavelet-like bases for the fast solution of second-kind integral equations, *SIAM J. Sci. Comput.* **14**(1), 159 (1993).
3. A. Dutt, M. Gu, and V. Rokhlin, Fast algorithms for polynomial interpolation, integration, and differentiation, *SIAM J. Numer. Anal.* **33**(5), (1996).
4. M. A. Epton and B. Dembart, Multipole translation theory for the three-dimensional Laplace and Helmholtz equations, *SIAM J. Sci. Comput.* **16**(4), 865 (1995).
5. I. S. Gradshteyn and I. M. Ryzhik, *Table of Integrals, Series, and Products*, 5th ed. (Academic Press, New York, 1994).
6. L. Greengard and V. Rokhlin, A fast algorithm for particle simulations, *J. Comput. Phys.* **73**(2), 325 (1987).
7. L. Greengard and V. Rokhlin, A new version of the fast multipole method for the Laplace equation in three dimensions, *Acta Numer.* 229 (1997).
8. V. H. Golub and C. H. Van Loan, *Matrix Computations* (Johns Hopkins Univ. Press, Baltimore, 1983).
9. T. Hrycak and V. Rokhlin, *An Improved Fast Multipole Algorithm for Potential Fields*, Research Report 1089, Computer Science Department, Yale, 1995.
10. R. Jakob-Chien and B. Alpert, A fast spherical filter with uniform resolution, *J. Comput. Phys.* **136**(2), 580 (1997).
11. S. Kapur and D. E. Long, IES^3: A fast integral equation solver for efficient 3-dimensional extraction, in *37th International Conference on Computer Aided Design, Nov. 1997.*
12. S. Kapur, D. E. Long, and J. Zhao, Efficient full-wave simulation in layered, lossy media, in *Proceedings of the IEEE Custom Integrated Circuits Conference, May 1998.*
13. S. A. Orszag, Fourier series on spheres, *Mon. Weather Rev.* **102**, 56 (1974).
14. J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, 2nd ed. (Springer-Verlag, New York/Berlin, 1993).
15. N. Yarvin and V. Rokhlin, An improved fast multipole algorithm for potential fields on the line, *SIAM J. Numer. Anal.*, to appear.

Waves Random Media **9** (1999) 441–457. Printed in the UK PII: S0959-7174(99)95309-0

An improved operator expansion algorithm for direct and inverse scattering computations

R Coifman†, M Goldberg‡, T Hrycak§, M Israeli|| and V Rokhlin§

† Department of Mathematics, Yale University, New Haven, CT 06520, USA
‡ Physical Sciences Department, York College of Pennsylvania, York, PA 17405, USA
§ Department of Computer Science, Yale University, New Haven, CT 06520, USA
|| Department of Computer Science, Technion, Haifa, Israel

Received 18 June 1998, in final form 9 February 1999

Abstract. In the first part of the paper we present an implementation of Milder's operator expansion formalism for acoustic scattering from a rough non-periodic surface. Our main contribution to the forward-field calculation is the development of two accurate ways of computing the order-zero normal differentiation operator N_0. The accuracy of our implementation is tested numerically. In the second part of our paper we apply this approach, combined with a continuation method, to an inverse scattering problem. The resulting scheme performs significantly better than the classical first-order methods.

1. Introduction

Scattering theory has been an active area of research for several decades. Several related problems belong to this field: acoustic and electromagnetic scattering form two large classes, which are further subdivided by assumptions on the underlying media and on the boundary conditions.

In direct problems one wants to calculate the field scattered by a given object. In two common situations, one knows either the values of the field on the scatterer (the Dirichlet problem), or the values of the normal derivative of the field on the boundary (the Neumann problem). Direct problems are usually well posed.

Inverse problems involve reconstructing the shape of a scatterer from the scattered field. These problems are ill posed: the solution has an unstable dependence on the input data.

For the convenience of the reader, we shall outline the progress made in acoustic scattering in a homogeneous medium from a sound-soft obstacle. A thorough discussion of this and related problems can be found in the references listed in the bibliography. The list of references is meant to be representative, rather than comprehensive.

The sound-soft scattering problem is characterized by the condition that the total field vanishes on the boundary of the scatterer. Thus, acoustic scattering is equivalent to the Dirichlet boundary value problem for the Helmholtz operator, with the scattered field equal to the negative of the known incident field. This problem is frequently solved by methods of potential theory. The single- and double-layer potentials relate a charge density on the boundary of the scatterer to the limiting values of the field and its normal derivative. The resulting integral equation is then solved in an appropriate function space, a common choice being the Lebesgue space L^2.

0959-7174/99/030441+17$19.50 © 1999 IOP Publishing Ltd

If the boundary is sufficiently smooth (C^2, for example) the method of layer potentials falls within the scope of Fredholm theory, (see [3]). When the boundary is merely Lipschitz, the Dirichlet problem becomes much more difficult and was first studied for the Laplace operator, corresponding to a zero wavenumber. The boundedness of the double-layer potential as an operator on L^2 is a deep result in real-variable theory, proved in [1] for arbitrary Lipschitz constants (see also [2] for a survey of related topics). Invertibility of the double-layer potential in L^2 was first proved in [17], and extended to other L^p spaces in [6]. A thorough description of related research, together with an extensive bibliography, is given in [9]. Extensions to non-zero wavenumbers and higher dimensions are obtained and described in [7, 11, 14, 15], ([14] has an extensive bibliography).

For the direct problem, a straightforward numerical solution of the integral equations for the scattered field leads to an $O(n^3)$ algorithm.

For the inverse problem, numerical methods must cope with the problem's inherent ill posedness. Some commonly used approaches require that the scattered field can be analytically continued across the boundary of the scatterer, which makes the problem even more unstable. References [4, 10] contain detailed descriptions of these methods and discuss the difficulties associated with them.

In this paper, we consider both the direct and inverse problems of acoustic scattering in a homogeneous medium. Following Milder [12, 13], we start from the boundary integral equation formulation and expand the scattering amplitude in a series of readily computable terms. The principal tool in this formalism is the admittance operator relating the scattered field and its normal derivative at the scattering surface. See [18] for a thorough discussion of the operator expansion method and other issues in rough surface scattering.

We adapt Milder's theory to fast numerical evaluation of the field scattered from rough (Lipschitz) surfaces with compact support. Other authors, see [8], have already reported numerical implementations of Milder's theory. Our contribution, in the case of forward-scattering computations, is to implement N_0 (the order-zero normal differentiation operator) accurately, for the case of a compact boundary. We resolve the problems caused by the singularity of the symbol of N_0 as a pseudo-differential operator and that of the associated integral kernel. We also implement N_2. In two dimensions, the results of our implementations are compared with the exact solution obtained by classical integral-equation methods. We have validated our method numerically for boundaries with Lipschitz constant less than $\frac{1}{10}$. In the second part of the paper, we approximate N_s, the inversion-symmetric form of the admittance operator, by N_0 in the forward-field equation and invert the resulting expression to solve an inverse scattering problem in the far-field regime. We use a continuation method with respect to the frequency: at each step we apply Newton's method with the starting point given by the output from the previous step. Thus at each stage we create an approximation to the curve filtered at a higher frequency. Our method recovers some nonlinear effects not accounted for by the classical Fourier inversion method, and works well in some situations where the linear term approximation fails completely.

The paper is organized as follows. Section 2 introduces the notation used in the paper. Section 3 contains a detailed description of Milder's formalism, as well as the algebraic transformations to ensure that the relevant operators always act on functions of compact support. Then we describe two implementations of the operator N_0 and compare them. The section concludes with numerical results for the forward-field computations. We consider an inverse scattering problem in section 4 and discuss our continuation method for solving it. This section also includes some numerical experiments in surface reconstruction. We conclude with a summary in section 5.

2. Notation and definitions

We shall associate with the vector $X = (x_1, x_2, x_3) \in \mathbb{R}^3$, the vector $\bar{X} = (x_1, x_2, -x_3)$. x without subscripts will denote a vector in $\mathbb{R}^2$ and we shall sometimes write X as (x, x_3). Our scattering surface is denoted by Γ and is given by the graph of a compactly supported Lipschitz function $\zeta : \mathbb{R}^2 \to \mathbb{R}$. The points on the surface are thus of the form $(x, \zeta(x))$. The free-space Green's function $G(X, Y)$ for the wavenumber k is given by the formula

$$G(X, Y) = \frac{1}{4\pi} \frac{\exp[ik|X - Y|]}{|X - Y|} \tag{1}$$

for $X \neq Y$.

We shall frequently denote $G(X, Y)$ by $G_X(Y)$. We shall also use the following expression for G:

$$G((x, z), (x_0, z_0)) = g(|(x, z) - (x_0, z_0)|) \tag{2}$$

where $(x, z) \neq (x_0, z_0)$ and

$$g(r) = \frac{1}{4\pi} \frac{e^{ikr}}{r}. \tag{3}$$

Functions satisfying the Helmholtz equation will be called metaharmonic.

3. Computation of the scattered field

We consider the Dirichlet problem for acoustic scattering from a compactly supported perturbation of the plane. In subsection 3.1, we describe Milder's operator expansion formalism. We also discuss a modification we make to ensure that all integrations are performed over compact regions. The next two subsections (3.2 and 3.3) form the main part of our contribution to the forward-scattering computations: two implementations of the order-zero normal differentiation operator N_0. Because of the central role N_0 plays in the expansion formalism, we feel it is of interest to describe different ways of implementing it. In subsection 3.4, we compare the two methods. The last subsection (3.5) presents some numerical examples of computations of the scattered field.

3.1. The operator expansion formalism

The surface Γ of the scatterer is given by the graph of a compactly supported Lipschitz function $\zeta : \mathbb{R}^2 \to \mathbb{R}$. We consider the Dirichlet problem for the Helmholtz equation, i.e. we wish to solve

$$(\Delta + k^2)\Phi_{\text{scat}} = 0 \tag{4}$$

in the region lying above Γ, with the sound-soft boundary condition

$$\Phi_{\text{scat}}|_\Gamma = -\Phi_{\text{inc}}|_\Gamma \tag{5}$$

where Φ_{inc} is the (known) incoming wave and Φ_{scat} is the scattered wave.

Following Milder, see [12, 13], we begin with the Green–Helmholtz integral for the scattered field:

$$\Phi_{\text{scat}}(R) = \int_\Gamma \left(\frac{\partial G_R}{\partial n}(X)\Phi_{\text{scat}}(X) - \frac{\partial \Phi_{\text{scat}}}{\partial n}(X) G_R(X) \right) \mathrm{d}s(X) \tag{6}$$

where the free-space Green's function is defined by

$$G_R(X) = \frac{\exp[ik|X - R|]}{4\pi|X - R|}. \tag{7}$$

Milder has modified this formula to obtain

$$\Phi_{\mathrm{scat}}(R) = 2 \int_{\mathbb{R}^2} G_R(y, \zeta(y))(N_s \Phi_{\mathrm{inc}})(y)\, \mathrm{d}y \tag{8}$$

where N_s has a formal operator power series expansion in ζ. Only even powers of ζ occur in the expansion, and N_s can be written as a series of operators

$$N_s = \sum_{j=0}^{\infty} N_{2j} = N_0 + N_2 + \cdots . \tag{9}$$

Already, the first two terms of this expansion provide an order-four approximation to the scattered potential, which surpasses the classical ones of Bragg or Kirchhoff (see [12]). The expressions for the operators N_0 and N_2 are given by the following formulae:

$$N_0 f = \left(\mathrm{i}\sqrt{k^2 - |\eta|^2}\hat{f}(\eta)\right)^{\vee} \tag{10}$$

$$N_2 f = -\tfrac{1}{2} N_0[\zeta, [\zeta, N_0]] N_0 f \tag{11}$$

where

$$[\zeta, N_0]g = \zeta(N_0 g) - N_0(\zeta g) \tag{12}$$

$\hat{f}$ is the Fourier transform and $\check{f}$ is the inverse Fourier transform of f.

Higher-order terms have simple expressions in terms of higher-order commutators, although their implementation gradually becomes more difficult.

Alternatively, N_0 can be viewed as a convolution operator with kernel $K(x, y)$ given by

$$K(x, y) = -2 \frac{g'(|x - y|)}{|x - y|} \tag{13}$$

where

$$g(r) = \frac{1}{4\pi} \frac{\mathrm{e}^{ikr}}{r}. \tag{14}$$

Note, that the kernel $K(x, y)$ is singular and is not a rapidly decaying function of $|x - y|$. Any accurate numerical implementation has to overcome these problems.

In our experiments the incident field originates at a point source located at S, so that

$$\Phi_{\mathrm{inc}}(Y) = G_S(Y). \tag{15}$$

We calculate the scattered field $\Phi_{\mathrm{scat}}(R)$ using N_0 or $N_0 + N_2$ instead of N_s. The resulting approximations are correct through second and fourth order in ζ, respectively. However, one cannot use formula (8) directly, since the functions $N_0\Phi_{\mathrm{inc}}$, $(N_0 + N_2)\Phi_{\mathrm{inc}}$ and $G_R(y, \zeta(y))$ are supported on the whole plane. Therefore, we modify formula (8) so that all non-local operators are applied to compactly supported functions and the final integration is performed on a compact set. First, since $G_{\bar{S}}(y)$ is metaharmonic above the boundary, (8) applied to $G_{\bar{S}}(y)$ gives:

$$G_{\bar{S}}(R) = -2 \int G_R(y, \zeta(y)) N_s G_{\bar{S}}(y)\, \mathrm{d}y \tag{16}$$

where $\bar{S}$ is the reflection of S across the XY-plane. Combining (15), (16) with (8), we obtain

$$\Phi_{\mathrm{scat}}(R) = -G_{\bar{S}}(R) + 2 \int G_R(y, \zeta(y)) N_s (G_S - G_{\bar{S}})(y)\, \mathrm{d}y. \tag{17}$$

Note that the difference $G_S - G_{\bar{S}}$ vanishes outside the support of ζ.

Even though $G_S - G_{\bar{S}}$ is compactly supported, $N_s(G_S - G_{\bar{S}})$, in general, is not. We shall now describe the additional modifications that are made to (17) after N_s is replaced by N_0, to ensure integration over a compact set. Defining

$$\Phi^0_{\text{scat}}(R) = -G_{\bar{S}}(R) + 2\int G_R(y, \zeta(y))N_0(G_S - G_{\bar{S}})(y)\,\mathrm{d}y \tag{18}$$

we have

$$\begin{aligned}\Phi^0_{\text{scat}}(R) = -G_{\bar{S}}(R) &+ 2\int G_R(y, 0)N_0(G_S - G_{\bar{S}})(y)\,\mathrm{d}y \\ &+2\int (G_R(y, \zeta(y)) - G_R(y, 0))N_0(G_S - G_{\bar{S}})(y)\,\mathrm{d}y.\end{aligned} \tag{19}$$

Since N_0 is a symmetric operator, and

$$N_0 G_R(y) = N_0 G_{\bar{R}}(y) = \frac{\partial G_{\bar{R}}}{\partial y_3}(y, 0) \tag{20}$$

we immediately obtain

$$\begin{aligned}\Phi^0_{\text{scat}}(R) = -G_{\bar{S}}(R) &+ 2\int \frac{\partial G_{\bar{R}}}{\partial y_3}(y, 0)(G_S - G_{\bar{S}})(y)\,\mathrm{d}y \\ &+2\int (G_R(y, \zeta(y)) - G_R(y, 0))N_0(G_S - G_{\bar{S}})(y)\,\mathrm{d}y.\end{aligned} \tag{21}$$

Since both $G_R(y, \zeta(y)) - G_R(y, 0)$ and $\partial G_{\bar{R}}/\partial y_3$ are compactly supported, we see that the evaluation of $\Phi^0_{\text{scat}}(R)$ can be reduced to evaluation of inner products of the form $\langle N_0 f, g\rangle = \int N_0 f(y) g(y)\,\mathrm{d}y$, where both f and g are compactly supported.

The operator N_2 requires several similar decompositions starting from (17). We omit the details.

3.2. *Implementation of the operator N_0*

As shown in the previous subsection, computation of the approximate scattered field can be reduced to evaluation of inner products of the form $\langle N_0 f, g\rangle$, where both f and g are compactly supported.

A straightforward numerical implementation of N_0 would consist of approximating the Fourier integral by a DFT, multiplying by the symbol of N_0, and then applying an approximate inverse Fourier transform via another DFT. However, the symbol of N_0 as a pseudo-differential operator, $\mathrm{i}\sqrt{k^2 - |\eta^2|}$, is not differentiable on the circle $|\eta| = k$. Therefore, this direct approach would result in a low-order integration scheme and require a very fine uniform discretization in frequency to give accurate results.

In this subsection, we demonstrate one way of resolving this problem. Our approach can be applied to compute other Fourier integral operators with singular kernels. In our numerical experiments, we approximate Lipschitz curves and surfaces by smooth functions. Thus the function f (and g) is smooth in addition to being compactly supported. Therefore, the function $\hat{f}$ is numerically compactly supported and integrations involving products of $\hat{f}$ are effectively on compact subsets of the frequency space.

Our method of computing $\langle N_0 f, g\rangle$ involves expressing N_0 as a sum of two operators, T_1 and T_2, with the following properties:

- the symbol of T_1 is continuously differentiable to a prescribed order, and
- T_2 is a convolution with a smooth function.

We evaluate T_1 using the FFT on the frequency side. Since the symbol of T_1 is several times differentiable, it can be sampled relatively coarsely and still yield a good approximation.

The convolution with the smooth kernel of T_2 can be implemented efficiently by an FFT, where this time the FFT is not viewed as a discretization of the continuous Fourier transform, as it was when evaluating T_1, but as an algebraic operation which diagonalizes the discrete convolution. $\langle N_0 f, g\rangle$ is then evaluated by integration over the compact support of g.

We shall exhibit the decomposition of N_0 in three dimensions, the result being valid in two dimensions with only minor modifications.

We note (see [13]), that

$$N_0 f(x) = \frac{1}{(2\pi)^2}\int_{\mathbb{R}^2} \mathrm{i}q(\eta)\mathrm{e}^{\mathrm{i}x\cdot\eta}\hat{f}(\eta)\,\mathrm{d}\eta \tag{22}$$

where $q(\eta) = \sqrt{k^2 - |\eta|^2}$ is chosen to have a positive imaginary part when $|\eta|^2 > k^2$.

We fix a positive integer m and a positive real x_3. We decompose $N_0 f$ into two terms:

$$\begin{aligned}N_0 f(x) &= T_1 f(x) + T_2 f(x)\\ &= \frac{1}{(2\pi)^2}\int_{\mathbb{R}^2} \mathrm{i}q(\eta)[1-\mathrm{e}^{\mathrm{i}q(\eta)x_3}]^m \mathrm{e}^{\mathrm{i}x\cdot\eta}\hat{f}(\eta)\,\mathrm{d}\eta\\ &\quad+\frac{1}{(2\pi)^2}\int_{\mathbb{R}^2} \mathrm{i}q(\eta)\{1-[1-\mathrm{e}^{\mathrm{i}q(\eta)x_3}]^m\}\mathrm{e}^{\mathrm{i}x\cdot\eta}\hat{f}(\eta)\,\mathrm{d}\eta.\end{aligned} \tag{23}$$

Let us first look at T_1. Its symbol, $\sigma(T_1)$, is given by

$$\begin{aligned}\sigma(T_1) &= \mathrm{i}q(\eta)[1-\mathrm{e}^{\mathrm{i}q(\eta)x_3}]^m\\ &= \mathrm{i}q(\eta)\left[-\mathrm{i}q(\eta)x_3 + \frac{q^2(\eta)x_3{}^2}{2} + \ldots\right]^m\\ &= c_1 q^{m+1}(\eta) + c_2 q^{m+2}(\eta) + \ldots.\end{aligned} \tag{24}$$

If m is odd, then $m+1$ is even, and $q^{m+1}(\eta)$ is a polynomial. Now, for $j = 1, 2$,

$$\frac{\mathrm{d}}{\mathrm{d}\eta_j}q(\eta) = \frac{\mathrm{d}}{\mathrm{d}\eta_j}\left(k^2 - |\eta|^2\right)^{1/2} = \frac{c\eta_j}{q(\eta)} \tag{25}$$

and

$$\frac{\mathrm{d}}{\mathrm{d}\eta_j}q^l(\eta) = cq^{l-2}(\eta)\eta_j. \tag{26}$$

Thus, each derivative in η reduces the exponent of q by two. If $l = 2j+1$, then $q^l(\eta)$ is j times continuously differentiable. In the above, if $m = 2n+1$, $m+2 = 2(n+1)+1$, then $\sigma(T_1)$, the symbol of T_1, is $n+1$ times continuously differentiable.

As for the operator T_2, we write

$$T_2(f)(x) = \int_{\mathbb{R}^2} K(x-y)f(y)\,\mathrm{d}y. \tag{27}$$

One can show that

$$K(x) = \sum_{n=1}^{m}(-1)^{n+1}\binom{m}{n}h(k, x, nx_3) \tag{28}$$

where

$$\begin{aligned}h(k, x, x_3) = -2\frac{\exp[\mathrm{i}k\sqrt{x^2+x_3^2}]}{4\pi\sqrt{x^2+x_3^2}}&\Big\{\mathrm{i}k(x^2+x_3^2)^{-1/2} - (k^2x_3^2+1)(x^2+x_3^2)^{-1}\\ &-3\mathrm{i}kx_3^2(x^2+x_3^2)^{-3/2} + 3x_3^2(x^2+x_3^2)^{-2}\Big\}.\end{aligned} \tag{29}$$

Moreover, $h(k, x, x_3)$ is a smooth function of x for a positive x_3, and thus $K(x)$ is also smooth. Details of the derivation are given in the appendix.

3.3. An alternative implementation of the operator N_0

There is an alternative way of implementing the operator N_0. We can regard N_0 as a convolution with an integral kernel, which has a singularity at zero. This section sketches the details of this approach. The interested reader may see [16] for a thorough discussion of the relevant issues. In the following we derive an explicit expression for the kernel.

The Green's function for the upper half-space $G_{\{z>0\}}$ can be expressed in terms of the free-space Green's function G as follows,

$$G_{\{z>0\}}((x, z), (x_0, z_0)) = G((x, z), (x_0, z_0)) - G((x, -z), (x_0, z_0)). \tag{30}$$

The Poisson kernel p for the upper half-space is the outward normal derivative of the Green's function

$$\begin{aligned} p(x, (x_0, z_0)) &= -\frac{\partial}{\partial z} G_{\{z>0\}}((x, z), (x_0, z_0))\bigg|_{z=0} \\ &= 2g'(|(x, 0) - (x_0, z_0)|)\frac{z_0}{|(x, 0) - (x_0, z_0)|}. \end{aligned} \tag{31}$$

The Dirichlet-to-Neumann operator N_0 can be expressed by the formula

$$N_0 f(x) = \lim_{z\to 0} -\frac{\partial}{\partial z}\int_{\mathbb{R}^2} p(y, (x, z)) f(y)\,\mathrm{d}y. \tag{32}$$

The kernel $K(x, y)$ of the Dirichlet-to-Neumann operator N_0, for $x \neq y$, is therefore the outward normal derivative of the Poisson kernel p (see also [18]),

$$K(x, y) = -\frac{\partial}{\partial z} p(y, (x, z))\bigg|_{z=0} = -2\frac{g'(|x - y|)}{|x - y|}. \tag{33}$$

The operator N_0 has been implemented via the following approximation

$$\begin{aligned} N_0 f(x) \approx{}& \text{Trapezoidal sum for } \int K(x, y) f(y)\,\mathrm{d}y \\ &+ c_1 f(x) h^{-1} + c_2 \Delta f(x) h + c_3 f(x) k^2 h + \mathrm{O}(h^3) \end{aligned} \tag{34}$$

where Δ is the Laplace operator in $\mathbb{R}^2$ and h is the side-length of an elementary grid square. The constants c_1, c_2, c_3 can be computed numerically from the formula (34) using Richardson extrapolation, see [5], p 269.

A similar approach applies to the two-dimensional case. The free-space Green's function is then given by the formula

$$\rho(r) = \frac{\mathrm{i}}{4} H_0(kr) \tag{35}$$

and the kernel of N_0 is equal to

$$K(x, y) = -2\frac{\rho'(r)}{r} = \frac{\mathrm{i}k}{2}\frac{H_1(k|x - y|)}{|x - y|}. \tag{36}$$

We use the following approximation:

$$N_0 f(x) \approx \text{Trapezoidal sum for the } \int K(x, y) f(y)\,\mathrm{d}y + a_1(h) f(x) + a_2(h) f''(x) \tag{37}$$

where

$$a_1(h) = -\frac{\pi}{3h} - \frac{1}{2\pi}\left(E - \frac{1}{2} + \log\left(\frac{hk}{4\pi}\right)\right) hk^2 - \frac{\zeta(3)}{4\cdot(2\pi)^3} h^3k^4 + \frac{\mathrm{i}}{4}hk^2$$
$$a_2(h) = \frac{h}{2\pi} + \frac{\zeta(3)}{(2\pi)^3} h^3k^2 \tag{38}$$

and $E = 0.577\,215\ldots$ is the Euler constant.

3.4. Comparison of the two methods

We have described two different methods of implementing N_0. The first one, expressing N_0 as a sum of T_1 and T_2, seems to be rather general and may prove useful for other integral operators. The main idea is that a non-decaying, singular symbol is broken into two parts: the first is non-decaying but smooth, while the second is singular but rapidly decaying at infinity. The first part can be applied on the frequency side with a relatively coarse discretization to functions with a fast decaying Fourier transform. Thus we can accurately evaluate $T_1 f$ when f is smooth. The second symbol is not applied on the frequency side, but as a convolution operator on the space side. Since this symbol is rapidly decaying, the convolution kernel is smooth and, again, a relatively coarse discretization can be used. Thus we can accurately evaluate $T_2 f$ when f is compactly supported.

The second method of implementing N_0 illustrates how to calculate a convolution with a kernel having a singularity at 0 numerically. The method is more direct, but the correction coefficients have to be computed for each particular kernel.

3.5. Numerical results

In this subsection we present examples of numerical computations of approximate scattered fields. We report our results in two dimensions and compare them with the accurate values obtained using the classical integral-equation approach. We used the two-dimensional version of formula (18) to calculate $\Phi^0_{\mathrm{scat}}(R)$, and a similar expression when N_s is replaced by $N_0 + N_2$. The results have been obtained with N_0 implemented by the method described in section 3.3, after verifying that both methods give nearly identical results in test cases.

The integral-equation method requires, however, that the scatterer be bounded. When the scatterer is defined by a non-negative, compactly supported function ζ, it is possible to reduce the Dirichlet problem on the open domain above ζ to the Dirichlet problem for the exterior of a bounded region. To this end, we first construct a solution u to the Dirichlet problem for the upper half-space. The boundary values of u should match the given data away from the support of the curve and can be chosen arbitrarily on the support. Next we consider the lens-shaped region formed by reflecting ζ about the plane $z = 0$, and the antisymmetric Dirichlet boundary conditions given as follows: the boundary values on the upper half of the region are equal to the original ones minus the values of u on the curve, while the boundary values on the lower half are the negatives of the corresponding values on the upper half. We now solve the Dirichlet problem for the resulting symmetric domain with antisymmetric boundary values. Note that the solution vanishes everywhere on the plane $z = 0$ outside the bounded region. The sum of u and the solution for the symmetric region is the solution to the original problem.

Tables 1–3 present results of numerical simulations for a simple test curve. In all cases, the relative errors are computed for the reduced potential $\Phi = \Phi_{\mathrm{scat}} + G_{\bar{\zeta}}(R)$. Using the full potential, the relative errors are much smaller, but less meaningful. The errors are computed

Table 1. Relative error of the reduced potential with $N_s \approx N_0$.

	Height				
Wavenumber	1	0.5	0.25	0.125	0.0625
π	6.72×10^{-1}	1.74×10^{-1}	4.77×10^{-2}	1.27×10^{-2}	3.27×10^{-3}
2π	8.10×10^{-1}	3.24×10^{-1}	8.56×10^{-2}	2.20×10^{-2}	5.60×10^{-3}
4π	9.52×10^{-1}	3.92×10^{-1}	7.74×10^{-2}	1.85×10^{-2}	4.66×10^{-3}
8π	1.13×10^{0}	5.19×10^{-1}	9.43×10^{-2}	2.16×10^{-2}	5.05×10^{-3}
16π	1.24×10^{0}	4.82×10^{-1}	8.64×10^{-2}	2.21×10^{-2}	5.37×10^{-3}
32π	1.30×10^{0}	5.68×10^{-1}	8.34×10^{-2}	2.06×10^{-2}	5.49×10^{-3}

Table 2. Relative error of the reduced potential with $N_s \approx N_0 + N_2$.

	Height				
Wavenumber	1	0.5	0.25	0.125	0.0625
π	2.82×10^{-1}	2.21×10^{-2}	1.84×10^{-3}	1.34×10^{-4}	2.44×10^{-5}
2π	3.81×10^{-1}	2.10×10^{-2}	1.76×10^{-3}	1.25×10^{-4}	3.28×10^{-5}
4π	1.06×10^{0}	9.09×10^{-2}	5.67×10^{-3}	3.72×10^{-4}	5.32×10^{-5}
8π	7.81×10^{-1}	2.21×10^{-1}	9.81×10^{-3}	4.18×10^{-4}	7.59×10^{-5}
16π	1.04×10^{0}	3.64×10^{-1}	9.18×10^{-3}	4.47×10^{-4}	2.15×10^{-4}
32π	1.12×10^{0}	5.22×10^{-1}	7.98×10^{-3}	5.09×10^{-4}	6.76×10^{-4}

Table 3. Relative difference of the reduced potentials with $N_s \approx N_0$ and $N_s \approx N_0 + N_2$.

	Height				
Wavenumber	1	0.5	0.25	0.125	0.0625
π	8.59×10^{-1}	1.95×10^{-1}	4.94×10^{-2}	1.28×10^{-2}	3.28×10^{-3}
2π	8.68×10^{-1}	3.38×10^{-1}	8.69×10^{-2}	2.21×10^{-2}	5.62×10^{-3}
4π	9.86×10^{-1}	4.52×10^{-1}	8.21×10^{-2}	1.88×10^{-2}	4.68×10^{-3}
8π	1.03×10^{0}	5.80×10^{-1}	1.03×10^{-1}	2.20×10^{-2}	5.07×10^{-3}
16π	9.81×10^{-1}	6.54×10^{-1}	9.42×10^{-2}	2.25×10^{-2}	5.39×10^{-3}
32π	1.02×10^{0}	7.70×10^{-1}	9.04×10^{-2}	2.09×10^{-2}	5.48×10^{-3}

in the l^2 norm:

$$E = \frac{\left(\sum_i |\Phi_i - \tilde{\Phi}_i|^2\right)^{1/2}}{\left(\sum_i |\tilde{\Phi}_i|^2\right)^{1/2}} \tag{39}$$

where Φ_i is the reduced potential at the ith receiver obtained by the algorithm and $\tilde{\Phi}_i$ is the corresponding value obtained by solving the combined field integral equations directly (see [4], p 67, for a thorough description).

Note how the relative errors increase with the height of the curve, but that they remain nearly constant at a fixed height as the wavenumber increases.

Table 4 records the result of a scattering experiment performed for a curve having only low-frequency components. The objective was to determine the dependence of the term N_2 on the wavenumber of the incident field. We find that the error depends only weakly on the wavenumber of the incident field once it exceeds the highest frequency of the curve.

Table 1. Relative error of the reduced potential with $N_s \approx N_0$.

Wavenumber	Height 1	 0.5	 0.25	 0.125	 0.0625
π	6.72×10^{-1}	1.74×10^{-1}	4.77×10^{-2}	1.27×10^{-2}	3.27×10^{-3}
2π	8.10×10^{-1}	3.24×10^{-1}	8.56×10^{-2}	2.20×10^{-2}	5.60×10^{-3}
4π	9.52×10^{-1}	3.92×10^{-1}	7.74×10^{-2}	1.85×10^{-2}	4.66×10^{-3}
8π	1.13×10^{0}	5.19×10^{-1}	9.43×10^{-2}	2.16×10^{-2}	5.05×10^{-3}
16π	1.24×10^{0}	4.82×10^{-1}	8.64×10^{-2}	2.21×10^{-2}	5.37×10^{-3}
32π	1.30×10^{0}	5.68×10^{-1}	8.34×10^{-2}	2.06×10^{-2}	5.49×10^{-3}

Table 2. Relative error of the reduced potential with $N_s \approx N_0 + N_2$.

Wavenumber	Height 1	 0.5	 0.25	 0.125	 0.0625
π	2.82×10^{-1}	2.21×10^{-2}	1.84×10^{-3}	1.34×10^{-4}	2.44×10^{-5}
2π	3.81×10^{-1}	2.10×10^{-2}	1.76×10^{-3}	1.25×10^{-4}	3.28×10^{-5}
4π	1.06×10^{0}	9.09×10^{-2}	5.67×10^{-3}	3.72×10^{-4}	5.32×10^{-5}
8π	7.81×10^{-1}	2.21×10^{-1}	9.81×10^{-3}	4.18×10^{-4}	7.59×10^{-5}
16π	1.04×10^{0}	3.64×10^{-1}	9.18×10^{-3}	4.47×10^{-4}	2.15×10^{-4}
32π	1.12×10^{0}	5.22×10^{-1}	7.98×10^{-3}	5.09×10^{-4}	6.76×10^{-4}

Table 3. Relative difference of the reduced potentials with $N_s \approx N_0$ and $N_s \approx N_0 + N_2$.

Wavenumber	Height 1	 0.5	 0.25	 0.125	 0.0625
π	8.59×10^{-1}	1.95×10^{-1}	4.94×10^{-2}	1.28×10^{-2}	3.28×10^{-3}
2π	8.68×10^{-1}	3.38×10^{-1}	8.69×10^{-2}	2.21×10^{-2}	5.62×10^{-3}
4π	9.86×10^{-1}	4.52×10^{-1}	8.21×10^{-2}	1.88×10^{-2}	4.68×10^{-3}
8π	1.03×10^{0}	5.80×10^{-1}	1.03×10^{-1}	2.20×10^{-2}	5.07×10^{-3}
16π	9.81×10^{-1}	6.54×10^{-1}	9.42×10^{-2}	2.25×10^{-2}	5.39×10^{-3}
32π	1.02×10^{0}	7.70×10^{-1}	9.04×10^{-2}	2.09×10^{-2}	5.48×10^{-3}

in the l^2 norm:

$$E = \frac{\left(\sum_i |\Phi_i - \tilde{\Phi}_i|^2\right)^{1/2}}{\left(\sum_i |\tilde{\Phi}_i|^2\right)^{1/2}} \tag{39}$$

where Φ_i is the reduced potential at the ith receiver obtained by the algorithm and $\tilde{\Phi}_i$ is the corresponding value obtained by solving the combined field integral equations directly (see [4], p 67, for a thorough description).

Note how the relative errors increase with the height of the curve, but that they remain nearly constant at a fixed height as the wavenumber increases.

Table 4 records the result of a scattering experiment performed for a curve having only low-frequency components. The objective was to determine the dependence of the term N_2 on the wavenumber of the incident field. We find that the error depends only weakly on the wavenumber of the incident field once it exceeds the highest frequency of the curve.

From (43) we find that

$$G_S(y,\zeta(y)) - G_{\bar{S}}(y,\zeta(y)) = -\mathrm{i}\frac{\mathrm{e}^{\mathrm{i}kr}}{2\pi r}\exp[-\mathrm{i}k(\sigma_1,\sigma_2)\cdot y]\sin(k\sigma_3\zeta(y)) + \mathrm{O}\left(\frac{1}{r^2}\right). \tag{44}$$

Similarly,

$$G_R(y,\zeta(y)) - G_R(y,0) = \frac{\mathrm{e}^{\mathrm{i}kr}}{4\pi r}\exp[-\mathrm{i}k(\omega_1,\omega_2)\cdot y]\left(\mathrm{e}^{-\mathrm{i}k\omega_3\zeta(y)} - 1\right) + \mathrm{O}\left(\frac{1}{r^2}\right). \tag{45}$$

Moreover,

$$G_{\bar{R}}(y,y_3) = \frac{\mathrm{e}^{\mathrm{i}kr}}{4\pi r}\exp[-\mathrm{i}k\bar{\omega}\cdot(y,y_3)] + \mathrm{O}\left(\frac{1}{r^2}\right) \tag{46}$$

and therefore

$$\frac{\partial G_{\bar{R}}}{\partial y_3}(y,0) = \mathrm{i}k\omega_3\frac{\mathrm{e}^{\mathrm{i}kr}}{4\pi r}\exp[-\mathrm{i}k(\omega_1,\omega_2)\cdot y] + \mathrm{O}\left(\frac{1}{r^2}\right). \tag{47}$$

Combining (44), (45), (47) with (40), we obtain

$$\begin{aligned}\Phi_{\mathrm{scat}}(R) \approx{}& -G_{\bar{S}}(R) + k\omega_3\frac{\mathrm{e}^{2\mathrm{i}kr}}{4\pi^2r^2}\int_{\mathbb{R}^2}\exp[-\mathrm{i}k(\omega_1+\sigma_1,\omega_2+\sigma_2)\cdot y]\sin(k\sigma_3\zeta)\,\mathrm{d}y\\ &-\mathrm{i}\frac{\mathrm{e}^{2\mathrm{i}kr}}{4\pi^2r^2}\int_{\mathbb{R}^2}\exp[-\mathrm{i}k(\omega_1,\omega_2)\cdot y]\left(\mathrm{e}^{-\mathrm{i}k\omega_3\zeta} - 1\right)\\ &\times N_0\left(\exp[-\mathrm{i}k(\sigma_1,\sigma_2)\cdot y]\sin(k\sigma_3\zeta)\right)\mathrm{d}y + \mathrm{O}\left(\frac{1}{r^3}\right).\end{aligned} \tag{48}$$

This leads to an expression in terms of the Fourier coefficients

$$\begin{aligned}\Phi_{\mathrm{scat}}(R) \approx{}& -G_{\bar{S}}(R) + k\omega_3\frac{\mathrm{e}^{2\mathrm{i}kr}}{4\pi^2r^2}[\sin(k\sigma_3\zeta)]^\wedge(k\omega_1+k\sigma_1,k\omega_2+k\sigma_2))\\ &-\mathrm{i}\frac{\mathrm{e}^{2\mathrm{i}kr}}{4\pi^2r^2}\left[\left(\mathrm{e}^{-\mathrm{i}k\omega_3\zeta} - 1\right)N_0\left(\exp[-\mathrm{i}k(\sigma_1,\sigma_2)\cdot y]\sin(k\sigma_3\zeta)\right)\right]^\wedge(k\omega_1,k\omega_2)\\ &+\mathrm{O}\left(\frac{1}{r^3}\right).\end{aligned} \tag{49}$$

In the special case, when the source is directly above, this formula becomes

$$\begin{aligned}\Phi_{\mathrm{scat}}(R) \approx{}& -G_{\bar{S}}(R) + k\omega_3\frac{\mathrm{e}^{2\mathrm{i}kr}}{4\pi^2r^2}[\sin(k\zeta)]^\wedge(k\omega_1,k\omega_2)\\ &-\mathrm{i}\frac{\mathrm{e}^{2\mathrm{i}kr}}{4\pi^2r^2}\left[\left(\mathrm{e}^{-\mathrm{i}k\omega_3\zeta} - 1\right)N_0(\sin(k\zeta))\right]^\wedge(k\omega_1,k\omega_2) + \mathrm{O}\left(\frac{1}{r^3}\right).\end{aligned} \tag{50}$$

Similarly, for the two-dimensional case, one can derive the following formula:

$$\begin{aligned}\Phi_{\mathrm{scat}}(R) \approx{}& -G_{\bar{S}}(R) + \mathrm{i}\omega_3\frac{\mathrm{e}^{2\mathrm{i}kr}}{2\pi r}[\sin(k\zeta)]^\wedge(k\omega_1)\\ &+\frac{\mathrm{e}^{2\mathrm{i}kr}}{2\pi kr}\left[\left(\mathrm{e}^{-\mathrm{i}k\omega_3\zeta} - 1\right)N_0(\sin(k\zeta))\right]^\wedge(k\omega_1) + \mathrm{O}\left(\frac{1}{r^2}\right).\end{aligned} \tag{51}$$

Although we used expression (51) in our numerical experiments, we would like to mention the following formula because of its appealing simplicity. For small elevations $k\zeta$, the sines and the exponentials can be expanded in powers of their arguments, yielding

$$\Phi_{\mathrm{scat}}(R) \approx -G_{\bar{S}}(R) + \mathrm{i}k\omega_3\frac{\mathrm{e}^{2\mathrm{i}kr}}{2\pi r}(\zeta - \zeta N_0\zeta)^\wedge(k\omega_1) + \mathrm{O}\left(\frac{1}{r^2}\right). \tag{52}$$

A similar result holds in three dimensions.

Let us now describe the geometric setup in two dimensions. The function ζ is supported on the interval $[-1, 1]$. The receivers at which we measure the scattered field are located on a semicircle of radius 10^5 in such a way that their projections on the x-axis are equispaced. The number of receivers is $\lfloor 2k/\pi \rfloor$. The source is located at the point $(0, 10^5)$.

Our reconstruction of ζ proceeds as follows.

- **Step 0**. We set the initial approximation to zero.
- **Step 1**. We choose an initial value for the wavenumber k and seek an approximation to the function ζ by a trigonometric polynomial of degree not exceeding k. Substituting

$$\zeta = \sum_{n=-k}^{k} c_n e^{int} \tag{53}$$

  in (51), we solve for the coefficients c_n using Newton's method with the previous approximation as the starting point. The resulting solution represents the Fourier coefficients of ζ corresponding to the frequencies not exceeding k.
- **Step 2**. We increase k to a new value k' ($k' = 2k$ is a convenient choice). We repeat step 1 with the previous approximation to ζ as our starting point. More precisely, we approximate ζ by the Fourier series $\sum_{n=-k'}^{k'} c_n e^{int}$ and determine the coefficients c_n by solving (51) using Newton's method starting from the previous result:

$$c_n' = \begin{cases} c_n & \text{for } |n| \leqslant k \\ 0 & \text{for } |n| > k \end{cases} \tag{54}$$

  where the coefficients c_n come from step 1.

We now iterate step 1 and step 2 until we reach a prescribed frequency k_0. For a complete reconstruction we need to choose k_0 larger than the highest frequency of the curve.

We have observed experimentally that the continuation method described above converges for a larger class of surfaces than Newton's method starting at $\zeta = 0$.

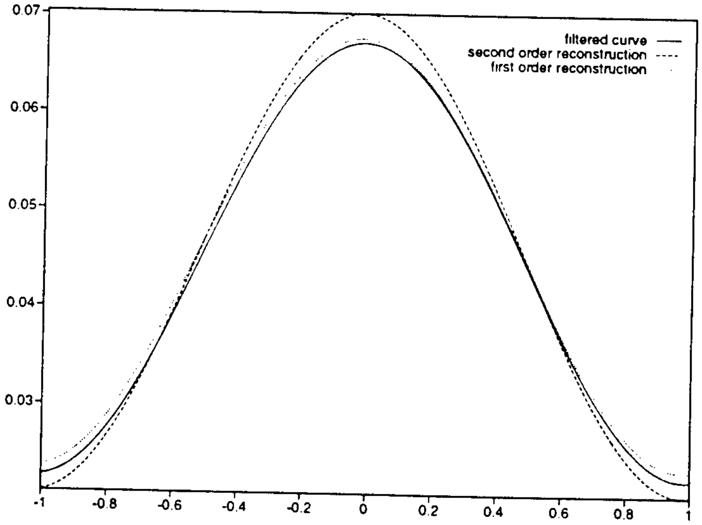


Figure 1. Reconstructions of the curve filtered at $k = \pi$. Filtered curve ——; second-order reconstruction - - - -; first-order reconstruction ······.

4.2. Numerical results

Figures 1–6 illustrate the continuation method as described in the previous subsection. The solid curve in the final figure is the unknown curve to be reconstructed. The first figure shows a filtered version of that curve at wavenumber π, and the reconstruction carried out using Newton's method starting from the zero curve. The second-order reconstruction is plotted together with the 'classical' linear reconstruction. The output of the second-order reconstruction is then the starting point for the next stage, where the wavenumber doubles (and so does the number of receivers on the semicircle). We proceed successively, as outlined in section 4.1, until we reach the wavenumber that is above the highest frequency of the curve. At each stage we attempt to reconstruct the true curve filtered at the corresponding wavenumber. The final reconstruction using the second-order method with continuation approximates the

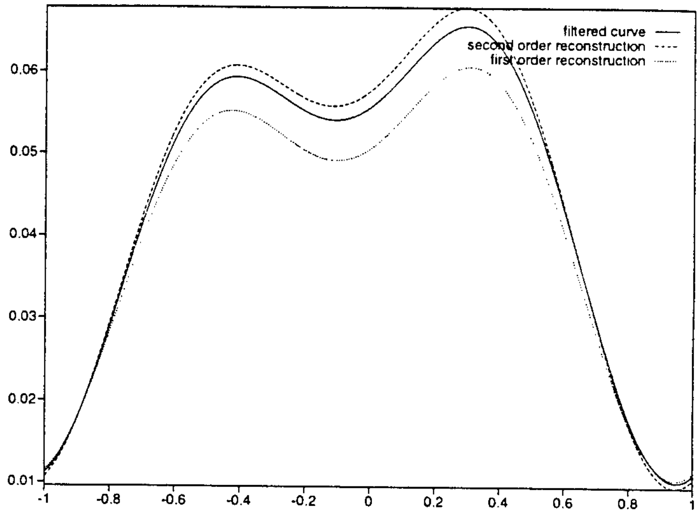


Figure 2. Reconstructions of the curve filtered at $k = 2\pi$. Filtered curve ——; second-order reconstruction - - - -; first-order reconstruction · · · · · ·.

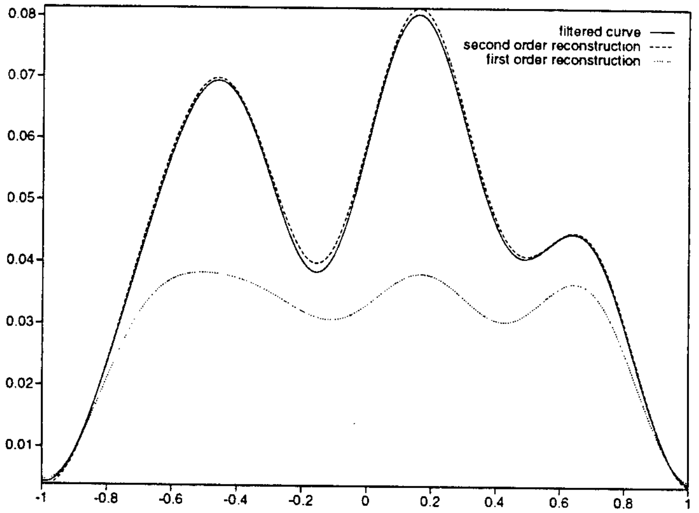


Figure 3. Reconstructions of the curve filtered at $k = 4\pi$. Filtered curve ——; second-order reconstruction - - - -; first-order reconstruction · · · · · ·.

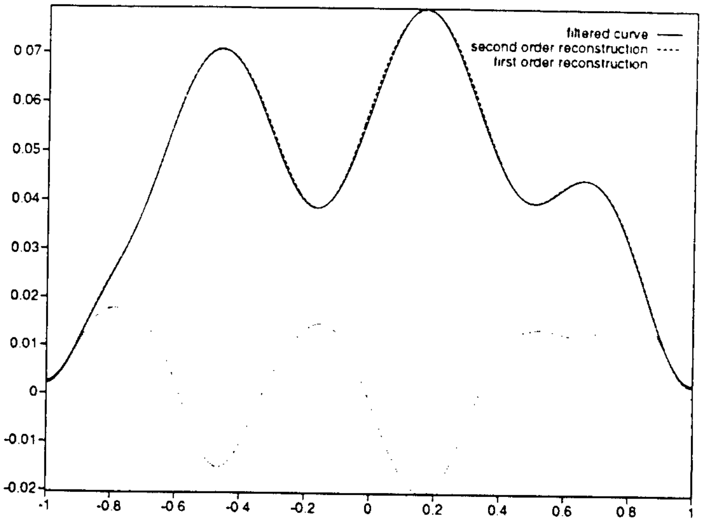


Figure 4. Reconstructions of the curve filtered at $k = 8\pi$. Filtered curve ——; second-order reconstruction - - - -; first-order reconstruction · · · · · ·.

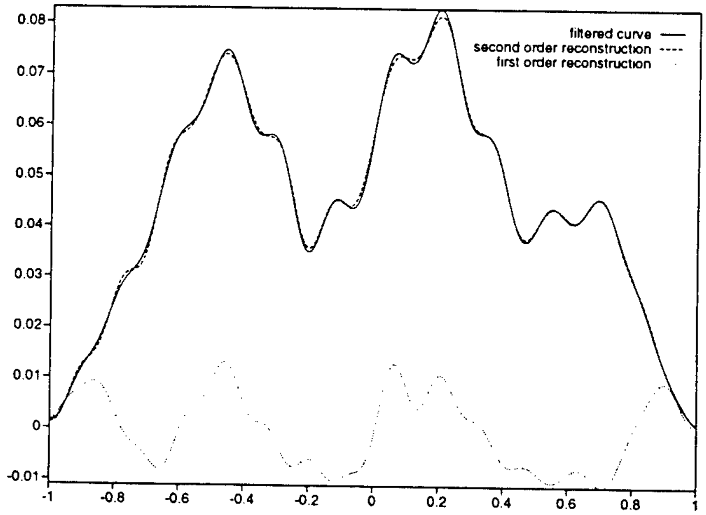


Figure 5. Reconstructions of the curve filtered at $k = 16\pi$. Filtered curve ——; second-order reconstruction - - - -; first-order reconstruction · · · · · ·.

curve very well. The first-order reconstruction is good for the first two stages but then moves further and further away from the actual curve.

5. Conclusions and summary

We present an implementation of Milder's operator expansion algorithm for acoustic scattering with Dirichlet boundary condition. We modify the integral used by Milder to ensure that all integral operators are applied to compactly supported functions and integrations are performed on bounded sets. Our main contribution to the forward-field calculation has been the development of two accurate ways of implementing the N_0 operator. We have also combined Milder's formalism together with a continuation method in frequency to reconstruct accurately rough boundaries with rather large heights. We have presented examples for which our

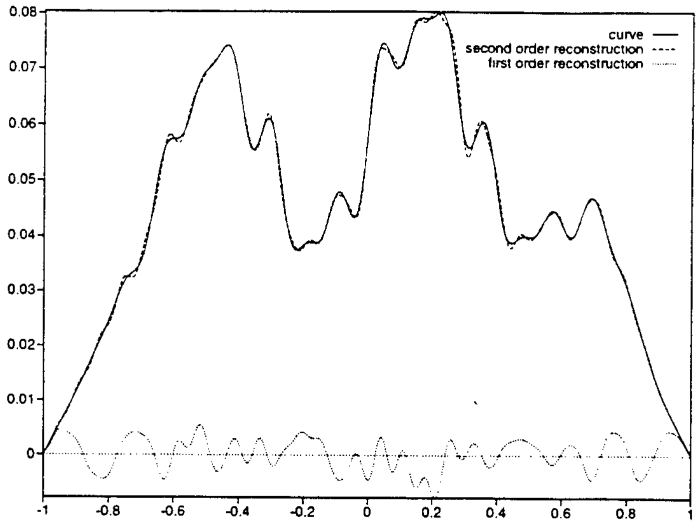


Figure 6. Reconstructions of the original curve with $k = 32\pi$. Original curve ——; second-order reconstruction - - - -; first-order reconstruction $\cdots\cdots$.

method using second-order terms works, but for which the first-order reconstruction fails. Our numerical results suggest that the higher-order approximation errors from incident fields having higher wavenumber than the frequency content of the boundary tend to remain nearly constant as the wavenumber of the incident field increases.

A scheme for the fast evaluation of the Helmholtz potentials can be added to accelerate the algorithm. Such methods are currently being developed by several authors.

Acknowledgments

The authors would like to thank Christopher Hatchell for editing the manuscript and the referees for their many helpful suggestions. This research was supported by DARPA/AFOSR under Grant F49620-97-1-0011.

Appendix

In this appendix, we provide a detailed derivation of the kernel of the convolution operator T_2 defined in section 3.2.

From (23) we obtain

$$\begin{aligned} T_2(f)(x) &= \frac{1}{(2\pi)^2}\int_{\mathbb{R}^2}\int_{\mathbb{R}^2} \mathrm{i}q(\eta)\{1-[1-\mathrm{e}^{\mathrm{i}q(\eta)x_3}]^m\}\mathrm{e}^{-\mathrm{i}y\cdot\eta}\mathrm{e}^{\mathrm{i}x\cdot\eta} f(y)\,\mathrm{d}y\,\mathrm{d}\eta \\ &= \int_{\mathbb{R}^2}\mathrm{d}y f(y)\frac{1}{(2\pi)^2}\int_{\mathbb{R}^2} \mathrm{i}q(\eta)\{1-[1-\mathrm{e}^{\mathrm{i}q(\eta)x_3}]^m\}\mathrm{e}^{\mathrm{i}(x-y)\cdot\eta}\,\mathrm{d}\eta \\ &= \int_{\mathbb{R}^2} K(x-y)f(y)\,\mathrm{d}y \end{aligned} \tag{55}$$

where

$$\begin{aligned} K(x) &= \frac{1}{(2\pi)^2}\int_{\mathbb{R}^2} \mathrm{i}q(\eta)\{1-[1-\mathrm{e}^{\mathrm{i}q(\eta)x_3}]^m\}\mathrm{e}^{\mathrm{i}x\cdot\eta}\,\mathrm{d}\eta \\ &= \frac{1}{(2\pi)^2}\int_{\mathbb{R}^2} \mathrm{i}q(\eta)\sum_{n=1}^{m}(-1)^{n+1}\binom{m}{n}\mathrm{e}^{\mathrm{i}q(\eta)nx_3}\mathrm{e}^{\mathrm{i}x\cdot\eta}\,\mathrm{d}\eta \end{aligned}$$

$$= \sum_{n=1}^{m} (-1)^{n+1} \binom{m}{n} h(k, x, nx_3) \tag{56}$$

with

$$h(k, x, x_3) \equiv \frac{1}{(2\pi)^2} \int_{\mathbb{R}^2} \mathrm{i}q(\eta) \mathrm{e}^{\mathrm{i}x\cdot\eta} \mathrm{e}^{\mathrm{i}q(\eta)x_3}\, \mathrm{d}\eta. \tag{57}$$

We note that $h(k, x, x_3)$ can also be expressed as

$$h(k, x, x_3) = \frac{-\mathrm{i}}{(2\pi)^2} \frac{\partial^2}{\partial x_3{}^2} \int_{\mathbb{R}^2} \mathrm{e}^{\mathrm{i}x\cdot\eta} \mathrm{e}^{\mathrm{i}q(\eta)x_3} \frac{\mathrm{d}\eta}{q(\eta)}. \tag{58}$$

We shall use the spectral form of the free-space Green's function, see [13],

$$\frac{\exp[\mathrm{i}k\|X - Y\|]}{4\pi\|X - Y\|} = \frac{\mathrm{i}}{2} \frac{1}{(2\pi)^2} \int_{\mathbb{R}^2} \exp[\mathrm{i}(x - y)\cdot\eta + \mathrm{i}q(\eta)|x_3 - y_3|] \frac{\mathrm{d}\eta}{q(\eta)}. \tag{59}$$

Again, since x_3 is positive, setting $Y = 0$, we obtain

$$\frac{\exp[\mathrm{i}k\sqrt{x^2 + x_3^2}]}{4\pi\sqrt{x^2 + x_3^2}} = \frac{\mathrm{i}}{2} \frac{1}{(2\pi)^2} \int_{\mathbb{R}^2} \exp[\mathrm{i}x\cdot\eta + \mathrm{i}q(\eta)x_3] \frac{\mathrm{d}\eta}{q(\eta)} \tag{60}$$

where $x^2 = x_1^2 + x_2^2$. Substitution of (60) into (58) gives

$$h(k, x, x_3) = -2\frac{\partial^2}{\partial x_3^2} \left(\frac{\exp[\mathrm{i}k\sqrt{x^2 + x_3^2}]}{4\pi\sqrt{x^2 + x_3^2}} \right). \tag{61}$$

After a straightforward calculation, we obtain:

$$h(k, x, x_3) = -2\, \frac{\exp[\mathrm{i}k\sqrt{x^2 + x_3^2}]}{4\pi\sqrt{x^2 + x_3^2}} \Big\{ \mathrm{i}k(x^2 + x_3^2)^{-1/2} - (k^2 x_3^2 + 1)(x^2 + x_3^2)^{-1} \\ -3\mathrm{i}k x_3^2 (x^2 + x_3^2)^{-3/2} + 3x_3^2 (x^2 + x_3^2)^{-2} \Big\}. \tag{62}$$

References

[1] Coifman R R, McIntosh A and Meyer Y 1982 L'intégrale de Cauchy définit un operateur borné sur L^2 pour les courbes Lipschitziennes *Ann. Math.* **116** 361–87 (in French)

[2] Coifman R R and Meyer Y 1986 *Nonlinear Harmonic Analysis, Operator Theory, and PDE (Ann. Math. Stud. 112)* (Princeton, NJ: Princeton University Press)

[3] Colton D and Kress R 1983 *Integral Equation Methods in Scattering Theory* (New York *et al* : Wiley)

[4] Colton D and Kress R 1998 *Inverse Acoustic and Electromagnetic Scattering Theory* 2nd edn (New York: Springer)

[5] Dahlquist G and Björck A 1974 *Numerical Methods* (Englewood Cliffs, NJ: Prentice-Hall)

[6] Dahlberg B and Kenig C 1987 Hardy spaces and the Neumann problem in L^p for Laplace's equation in Lipschitz domains *Ann. Math.* **125** 437–66

[7] Jaweth B and Mitrea M 1995 Higher-dimensional electromagnetic scattering theory on C^1 and Lipschitz domains *Am. J. Math.* **117** 929–63

[8] Kaczkowski P J and Thorsos E I 1994 Application of the operator expansion method to scattering from one-dimensional moderately rough Dirichlet random surfaces *J. Acoust. Soc. Am.* **96** 957–72

[9] Kenig C E 1994 *Harmonic Analysis Techniques for Second-Order Elliptic Boundary Value Problems (CBMS Regional Conference Series in Mathematics 83 (St Louis, 1991))* (Providence, RI: American Mathematical Society)

[10] Kirsch A 1996 *An Introduction to the Mathematical Theory of Inverse Problems* (New York: Springer)

[11] McIntosh A and Mitrea M 1996 Clifford algebras and Maxwell's equations in Lipschitz domains *Macquairie Mathematics Reports* No. 96/210

[12] Milder D M 1991 An improved formalism for wave scattering from rough surfaces *J. Acoust. Soc. Am.* **89** 529–41

[13] Milder D M 1996 Role of the admittance operator in rough-surface scattering *J. Acoust. Soc. Am.* **100** 759–68

[14] Mitrea D, Mitrea M and Pipher J 1997 Vector potential theory on non-smooth domains in $\mathbb{R}^3$ and applications to electromagnetic scattering *J. Fourier Anal. Appl.* **3** 131–92

[15] Mitrea M 1995 The method of layer potentials in electromagnetic scattering theory on non-smooth domains *Duke Math. J.* **77** 111–33

[16] Sidi A and Israeli M 1988 Quadrature methods for periodic singular and weakly singular Fredholm integral equations *J. Sci. Comput.* **3** 201–31

[17] Verchota G 1984 Layer potentials and regularity for the Dirichlet problem for Laplace's equation *J. Funct. Anal.* **59** 572–611

[18] Voronovich A G 1999 *Wave Scattering from Rough Surfaces* 2nd edn (New York: Springer)

SIAM J. NUMER. ANAL.
Vol. 37, No. 4, pp. 1138–1164

© 2000 Society for Industrial and Applied Mathematics

RAPID EVALUATION OF NONREFLECTING BOUNDARY KERNELS FOR TIME-DOMAIN WAVE PROPAGATION*

BRADLEY ALPERT†, LESLIE GREENGARD‡, AND THOMAS HAGSTROM§

Abstract. We present a systematic approach to the computation of exact nonreflecting boundary conditions for the wave equation. In both two and three dimensions, the critical step in our analysis involves convolution with the inverse Laplace transform of the logarithmic derivative of a Hankel function. The main technical result in this paper is that the logarithmic derivative of the Hankel function $H_\nu^{(1)}(z)$ of real order ν can be approximated in the upper half z-plane with relative error ε by a rational function of degree $d \sim O(\log|\nu| \log \frac{1}{\varepsilon} + \log^2|\nu| + |\nu|^{-1} \log^2 \frac{1}{\varepsilon})$ as $|\nu| \to \infty$, $\varepsilon \to 0$, with slightly more complicated bounds for $\nu = 0$. If N is the number of points used in the discretization of a cylindrical (circular) boundary in two dimensions, then, assuming that $\varepsilon < 1/N$, $O(N \log N \log \frac{1}{\varepsilon})$ work is required at each time step. This is comparable to the work required for the Fourier transform on the boundary. In three dimensions, the cost is proportional to $N^2 \log^2 N + N^2 \log N \log \frac{1}{\varepsilon}$ for a spherical boundary with N^2 points, the first term coming from the calculation of a spherical harmonic transform at each time step. In short, nonreflecting boundary conditions can be imposed to any desired accuracy, at a cost dominated by the interior grid work, which scales like N^2 in two dimensions and N^3 in three dimensions.

Key words. Bessel function, approximation, high-order convergence, wave equation, Maxwell's equations, nonreflecting boundary condition, radiation boundary condition, absorbing boundary condition

AMS subject classifications. 33C10, 41A20, 44A10, 44A35, 65D20

PII. S0036142998336916

1. Introduction.

A longstanding practical issue in numerical wave propagation and scattering problems concerns the reduction of an unbounded domain to a bounded domain by the imposition of nonreflecting boundary conditions at an artificial boundary. We restrict our attention to "time-domain" calculations, for which it is well-known that the exact nonreflecting conditions are global in both space and time. While the problem has been widely studied (see Givoli [1] for an overview), the boundary conditions used in practice typically introduce serious numerical artifacts. An exception is the method developed by Ting and Miksis [2], which relies on Kirchhoff's formula to solve the wave equation in an exterior domain, but which is computationally expensive. The two most common approaches are based on the construction of local differential boundary conditions [3, 4] or absorbing regions [5, 6], but neither provides a clear sequence of approximations which converge to the exact, nonlocal conditions. Recently, Sofronov [7] and, independently, Grote and Keller [8]

---

[*] Received by the editors April 7, 1998; accepted for publication (in revised form) May 20, 1999; published electronically March 23, 2000. Contribution of the U.S. Government not subject to copyright in the United States.

http://www.siam.org/journals/sinum/37-4/33691.html

[†] National Institute of Standards and Technology, 325 Broadway, Boulder, CO 80303 (alpert@boulder.nist.gov). This author was supported in part by DARPA under appropriation 9770400.

[‡] Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY 10012-1110 (greengar@cims.nyu.edu). This author was supported in part by U.S. Department of Energy under contract DE-FGO288ER25053 and DARPA/AFOSR under contract F94620-95-C-0075.

[§] Department of Mathematics and Statistics, University of New Mexico, Albuquerque, NM 87131 (hagstrom@ math.unm.edu). This author was supported in part by the U.S. Department of Energy under contract DE-FGO288ER25053, DARPA/ AFOSR under contract F94620-95-C-0075, and National Science Foundation under grant DMS-9600146.

have developed and implemented an integrodifferential approach for three-dimensional calculations using a spherical boundary and have demonstrated that high accuracy can be achieved at reasonable cost. In their schemes, the work is of the same order as the explicit finite difference or finite element calculation in the interior of the domain. For N^2 points on the spherical boundary, $O(N^3)$ work is required. Hagstrom and Hariharan [9] have shown that these conditions can be effectively implemented using only local operators, but at the cost of introducing a large number of auxiliary functions at the boundary. A somewhat more general, but closely related, integral formulation is introduced in [10, 11, 12]. The fundamental analytical tool in the latter papers is what we refer to as the *nonreflecting boundary kernel* which is the inverse Laplace transform of the logarithmic derivative of a Hankel function.

In this paper, we prove that the logarithmic derivative of a Hankel function can be approximated as a ratio of polynomials of modest degree, so that its inverse Laplace transform can be expressed as a sum of exponentials. Our analytical approach combines an extension of the Mittag–Leffler theorem with the approximation techniques of the fast multipole method. In particular, Theorem 4.1 presents an exact representation of the logarithmic derivative as a sum of poles plus a continuous density on the branch cut. Theorem 4.6, which is preceded by several technical lemmas, presents a reduced, approximate representation.

Using this approach, the cost of computing the nonreflecting boundary condition is comparable to that of a fast Fourier or spherical harmonic transform. For two-dimensional problems, $O(N\log N\log\frac{1}{\varepsilon})$ work is required at each time step, where N is the number of points used in the discretization of a cylindrical (circular) boundary. In three dimensions, the cost is proportional to $N^2\log^2 N + N^2\log N\log\frac{1}{\varepsilon}$ for a spherical boundary with N^2 points. The first term comes from the calculation of the spherical harmonic transform using the fast algorithm of [13, 14].

Other authors, including Nédélec [15] and Cruz and Sesma [16], have studied the logarithmic derivative of the Hankel function, based on a variety of techniques. In this paper we present a sum-of-poles representation for the logarithmic derivative of a Hankel function of real order ν bounded away from zero with accuracy ε for argument, z, satisfying $\mathrm{Im}(z)\geq 0$. The number of poles is bounded by $O\big(\log|\nu|\cdot\log\frac{1}{\varepsilon}+\log^2|\nu|+|\nu|^{-1}\log^2\frac{1}{\varepsilon}\big)$. A similar representation for $\nu=0$ is also derived which is valid for $\mathrm{Im}(z)\geq\eta>0$ requiring $O\big(\log\frac{1}{\eta}\cdot\log\frac{1}{\varepsilon}+\log\frac{1}{\varepsilon}\cdot\log\log\frac{1}{\varepsilon}+\log\frac{1}{\eta}\cdot\log\log\frac{1}{\eta}\big)$ poles.

In section 2, we introduce nonreflecting boundary kernels. In section 3 we collect background material in a form convenient for the subsequent development. Section 4 contains the analytical and approximate treatment of the logarithmic derivative, while a procedure for computing these representations is presented in Section 5. The results of our numerical computations are contained in section 6, and we present our conclusions in section 7.

2. Nonreflecting boundary kernels.

Let us first consider the wave equation

$$u_{tt}=c^2\nabla^2 u \tag{2.1}$$

in a two-dimensional annular domain $\rho_0<\rho<\rho_1$. The general solution can be expressed as

$$u(\rho,\phi,t)=\sum_{n=-\infty}^{\infty}e^{in\phi}\mathcal{L}^{-1}\big[a_n(s)\,K_n(\rho s/c)+b_n(s)\,I_n(\rho s/c)\big](t), \tag{2.2}$$

where K_n and I_n are modified Bessel functions (see, for example, [17, section 9.6]),

$$K_n(z) = \frac{\pi}{2} i^{n+1} H_n^{(1)}(z\, e^{\pi i/2}), \qquad I_n(z) = i^{-n} J_n(z\, e^{\pi i/2}), \qquad -\pi < \arg z \le \frac{\pi}{2}, \tag{2.3}$$

the coefficients a_n and b_n are arbitrary functions analytic in the right half-plane, $\mathcal{L}$ denotes the Laplace transform

$$\mathcal{L}[f](s) = \int_0^\infty e^{-st} f(t)\, dt, \tag{2.4}$$

and $\mathcal{L}^{-1}$ denotes the inverse Laplace transform

$$\mathcal{L}^{-1}[g](t) = \frac{1}{2\pi i} \int_{-i\infty}^{i\infty} e^{st} g(s)\, ds. \tag{2.5}$$

Likewise, for the wave equation in a three-dimensional domain $r_0 < r < r_1$, the general solution can be expressed as

$$u(r, \phi, \theta, t) = \sum_{n=-\infty}^{\infty} \sum_{m=-n}^{n} Y_{nm}(\phi, \theta)\, \mathcal{L}^{-1} \left[a_{nm}(s)\, \frac{K_{n+\frac{1}{2}}(rs/c)}{\sqrt{rs/c}} + b_{nm}(s)\, \frac{I_{n+\frac{1}{2}}(rs/c)}{\sqrt{rs/c}} \right](t). \tag{2.6}$$

If we imagine that $\rho = \rho_1$ (or $r = r_1$) is to be used as a nonreflecting boundary, then we can assume there are no sources in the exterior region and the coefficients $b_n(s)$ (or $b_{nm}(s)$) are zero. Let us now denote by $u_n(\rho, t)$ the function satisfying

$$\mathcal{L}[u_n](\rho, s) = a_n(s)\, K_n(\rho s/c). \tag{2.7}$$

Then

$$\begin{aligned} \mathcal{L}\left[\frac{\partial}{\partial \rho} u_n\right](\rho, s) &= a_n(s) \cdot \frac{s}{c} \cdot K_n'(\rho s/c) \\ &= \mathcal{L}[u_n](\rho, s) \cdot \left(\frac{s}{c} \frac{K_n'(\rho s/c)}{K_n(\rho s/c)} \right), \end{aligned} \tag{2.8}$$

so that

$$\frac{\partial}{\partial \rho} u_n(\rho, t) = u_n(\rho, t) * \mathcal{L}^{-1} \left[\frac{s}{c} \frac{K_n'(\rho s/c)}{K_n(\rho s/c)} \right](t), \tag{2.9}$$

where $*$ denotes Laplace convolution

$$(f * g)(t) = \int_0^t f(\tau)\, g(t - \tau)\, d\tau. \tag{2.10}$$

The convolution kernel in (2.9) is a generalized function. Its singular part is easily removed, however, by subtracting the first two terms of the asymptotic expansion

$$\frac{s}{c} \frac{K_n'(\rho s/c)}{K_n(\rho s/c)} \sim -\frac{s}{c} - \frac{1}{2\rho} + O(s^{-1}), \qquad s \to \infty. \tag{2.11}$$

From the assumption $u_n(\rho, t) = 0$ for $t \leq 0$ and standard properties of the Laplace transform we obtain the boundary condition

$$\frac{\partial}{\partial\rho}u_n(\rho,t) + \frac{1}{c}\frac{\partial}{\partial t}u_n(\rho,t) + \frac{1}{2\rho}u_n(\rho,t) = \int_0^t \sigma_n(t-\tau)\,u_n(\rho,\tau)\,d\tau, \tag{2.12}$$

where

$$\sigma_n(t) = \mathcal{L}^{-1}\left[\frac{s}{c} + \frac{1}{2\rho} + \frac{s}{c}\frac{K_n'(\rho s/c)}{K_n(\rho s/c)}\right](t), \tag{2.13}$$

which we impose at $\rho = \rho_1$.

Remark. The solution to the wave equation in physical space is recovered on the nonreflecting boundary from u_n by Fourier transformation:

$$u(\rho_1, \phi, t) = \sum_{n=-N/2}^{N/2-1} u_n(\rho_1, t)\,e^{in\phi}, \tag{2.14}$$

assuming N points are used in the discretization.

The analogous boundary condition in three dimensions is expressed in terms of the functions $u_{nm}(r,t)$ satisfying

$$\mathcal{L}[u_{nm}](r,s) = a_{nm}(s)\,\frac{K_{n+\frac{1}{2}}(rs/c)}{\sqrt{rs/c}}. \tag{2.15}$$

After some algebraic manipulation, assuming $u_{nm}(r,t) = 0$ for $t \leq 0$, we have

$$\frac{\partial}{\partial r}u_{nm}(r,t) + \frac{1}{c}\frac{\partial}{\partial t}u_{nm}(r,t) + \frac{1}{r}u_{nm}(r,t) = \int_0^t \omega_n(t-\tau)\,u_{nm}(r,\tau)\,d\tau, \tag{2.16}$$

where

$$\omega_n(t) = \mathcal{L}^{-1}\left[\frac{s}{c} + \frac{1}{2r} + \frac{s}{c}\frac{K_{n+\frac{1}{2}}'(rs/c)}{K_{n+\frac{1}{2}}(rs/c)}\right](t), \tag{2.17}$$

which we impose at $r = r_1$.

Note that the boundary conditions (2.12) and (2.16) are exact but nonlocal, since they rely on a Fourier (or spherical harmonic) transformation in space and are history dependent. The form of the history is simple, however, and expressed, for each separate mode, in terms of a convolution kernel which is the inverse Laplace transform of a function defined in terms of the logarithmic derivative of a modified Bessel function

$$\frac{d}{dz}\log K_\nu(z) = \frac{K_\nu'(z)}{K_\nu(z)}. \tag{2.18}$$

Remark. In three dimensions, the required logarithmic derivative of $K_{n+\frac{1}{2}}(z)$ is a ratio of polynomials, so that one can recast the boundary condition in terms of a differential operator of order n. The resulting expression would be equivalent to those derived by Sofronov [7] and Grote and Keller [8].

The remainder of this paper is devoted to the approximation of the logarithmic derivatives (2.18) as a ratio of polynomials of degree $O(\log\nu)$, from which the convolution kernels σ_n and ω_n can be expressed as a sum of decaying exponentials. This representation allows for the recursive evaluation of the integral operators in (2.12) and (2.16), using only $O(\log n)$ work per time step (see [18]). We note that, by Parseval's equality, the L_2 error resulting from convolution with an approximate kernel is sharply bounded by the L_∞ error in the approximation to the kernel's transform. Precisely, approximating the kernel $B(t)$ by the kernel $A(t)$ we find

$$\begin{aligned}\left\|A*u-B*u\right\|_2=\left\|\hat{A}\hat{u}-\hat{B}\hat{u}\right\|_2&\le\sup_{s\in i\mathbb{R}}\frac{|\hat{A}-\hat{B}|}{|\hat{B}|}\left\|\hat{B}\hat{u}\right\|_2\\&=\sup_{s\in i\mathbb{R}}\frac{|\hat{A}-\hat{B}|}{|\hat{B}|}\left\|B*u\right\|_2,\end{aligned}\tag{2.19}$$

where we assume that $\hat{A}$, $\hat{B}$, and $\hat{u}$ are all regular for $\mathrm{Re}(s)>0$. For finite times we may let s have a positive real part, η:

$$\left\|A*u-B*u\right\|_{L_2(0,T)}\le e^{\eta T}\sup_{s\in\eta+i\mathbb{R}}\frac{|\hat{A}-\hat{B}|}{|\hat{B}|}\left\|B*u\right\|_{L_2(0,T)}.\tag{2.20}$$

We therefore concentrate our theoretical developments on L_∞ approximations. For ease of computation, however, we compute our rational representations by least squares methods. These do generally lead to small relative errors in the maximum norm, as will be shown.

Since Hankel functions are more commonly used in the special function literature, we will write the logarithmic derivatives as

$$\frac{d}{dz}\log K_\nu(z)=\frac{d}{dz}\log H_\nu^{(1)}\left(z\,e^{\pi i/2}\right)=i\,\frac{H_\nu^{(1)'}\left(z\,e^{\pi i/2}\right)}{H_\nu^{(1)}\left(z\,e^{\pi i/2}\right)}.\tag{2.21}$$

We are, then, interested in approximating logarithmic derivative of the Hankel function on and above the real axis.

3. Mathematical preliminaries.

In this section we collect several well-known facts concerning the Bessel equation, the logarithmic derivative of the Hankel function, and pole expansions, in a form that will be useful in the subsequent analytical development.

3.1. Bessel's equation.

Bessel's differential equation

$$\frac{d^2u}{dz^2}+\frac{1}{z}\frac{du}{dz}+\left(1-\frac{\nu^2}{z^2}\right)u=0,\tag{3.1}$$

for $\nu\in\mathbb{R}$, has linearly independent solutions $H_\nu^{(1)}$ and $H_\nu^{(2)}$, known as Hankel's functions. These can be expressed by the formulae

$$H_\nu^{(1)}(z)=\frac{J_{-\nu}(z)-e^{-\nu\pi i}J_\nu(z)}{i\sin(\nu\pi)},\qquad H_\nu^{(2)}(z)=-\frac{J_{-\nu}(z)-e^{\nu\pi i}J_\nu(z)}{i\sin(\nu\pi)},\tag{3.2}$$

where the Bessel function of the first kind is defined by

$$J_\nu(z)=\left(\frac{z}{2}\right)^\nu\sum_{k=0}^{\infty}\frac{(-z^2/4)^k}{k!\Gamma(\nu+k+1)}.\tag{3.3}$$

The expressions in (3.2) are replaced by their limiting values for integer values of ν. (See, for example, [17, section 9.1].) For general ν, the functions $H_\nu^{(1)}$ and $H_\nu^{(2)}$ have a branch point at $z = 0$ and it is customary to place the corresponding branch cut on the negative real axis and impose the restriction $-\pi < \arg z \leq \pi$. We shall find it more convenient, however, to place the branch cut on the negative imaginary axis, with the restriction

$$-\frac{\pi}{2} \leq \arg z < \frac{3\pi}{2}. \tag{3.4}$$

Hankel's functions have especially simple asymptotic properties. In particular (see, for example, [19, section 7.4.1]),

$$H_\nu^{(1)}(z) \sim \left(\frac{2}{\pi z}\right)^{1/2} e^{i(z-\nu\pi/2-\pi/4)} \sum_{k=0}^{\infty} i^k \frac{A_k(\nu)}{z^k}, \tag{3.5}$$

$$H_\nu^{(1)\prime}(z) \sim \left(\frac{2}{\pi z}\right)^{1/2} e^{i(z-\nu\pi/2-\pi/4)} \sum_{k=0}^{\infty} i^k \frac{A_k(\nu)}{z^k}\left(-\frac{1}{2z} + i - \frac{k}{z}\right) \tag{3.6}$$

as $z \to \infty$, with $-\pi + \delta \leq \arg z \leq 2\pi - \delta$, where

$$A_k(\nu) = \frac{(4\nu^2 - 1^2)(4\nu^2 - 3^2)\cdots(4\nu^2 - (2k-1)^2)}{k!\,8^k}, \tag{3.7}$$

and the branch of the square root is determined by

$$z^{1/2} = e^{(\log|z| + i\arg z)/2}. \tag{3.8}$$

Finally we note the symmetry

$$H_\nu^{(1)}(z) = e^{-\nu\pi i} H_{-\nu}^{(1)}(z). \tag{3.9}$$

We also make use of the modified Bessel functions $K_\nu(z)$ and $I_\nu(z)$. These are linearly independent solutions of the equation obtained from (3.1) by the transformation $z \to iz$. Their Wronskian satisfies

$$K_\nu(z)I_\nu'(z) - K_\nu'(z)I_\nu(z) = z^{-1}. \tag{3.10}$$

Moreover we have for positive r [20]

$$H_\nu^{(1)}(re^{-i\pi/2}) = \frac{2}{\pi i} e^{-\nu\pi i/2}\left(e^{\nu\pi i}K_\nu(r) + \pi i I_\nu(r)\right). \tag{3.11}$$

Asymptotic expansions of $K_\nu(r)$ and $I_\nu(r)$ for r small and large are also known [17, sections 9.6 and 9.7]. For real r and $\nu \geq 0$ we have

$$K_\nu(r) \sim \begin{cases} \gamma - \log\dfrac{r}{2}, & \nu = 0, \\ \dfrac{\Gamma(\nu)}{2}\left(\dfrac{r}{2}\right)^{-\nu}, & \nu > 0, \end{cases} \qquad r \to 0, \tag{3.12}$$

$$I_\nu(r) \sim \frac{1}{\Gamma(\nu+1)}\left(\frac{r}{2}\right)^{\nu}, \qquad r \to 0, \tag{3.13}$$

$$K_\nu(r) \sim \sqrt{\frac{\pi}{2r}}\, e^{-r}, \qquad r \to \infty, \tag{3.14}$$

$$I_\nu(r) \sim \sqrt{\frac{1}{2\pi r}}\, e^{r}, \qquad r \to \infty. \tag{3.15}$$

Here $\gamma = 0.5772\ldots$ is the Euler constant.

Finally, we note the uniform expansions of Bessel functions for $\nu \to \infty$ given in [17]. For Hankel function and derivative we have

$$H_\nu^{(1)}(\nu z) \sim 2e^{-\pi i/3}\Big(\frac{4\zeta}{1-z^2}\Big)^{1/4}\frac{\mathrm{Ai}\big(e^{2\pi i/3}\nu^{2/3}\zeta\big)}{\nu^{1/3}}. \tag{3.16}$$

$$H_\nu^{(1)'}(\nu z) \sim \frac{4e^{-2\pi i/3}}{z}\Big(\frac{4\zeta}{1-z^2}\Big)^{-1/4}\frac{\mathrm{Ai}'\big(e^{2\pi i/3}\nu^{2/3}\zeta\big)}{\nu^{2/3}} \tag{3.17}$$

as $\nu \to \infty$, where we restrict z to $|\arg(z)| \le \pi/2$ and define

$$\frac{2}{3}\zeta^{3/2} = \log\frac{1+\sqrt{1-z^2}}{z} - \sqrt{1-z^2}. \tag{3.18}$$

Here, $\mathrm{Ai}(t)$ denotes the Airy function [17, section 10.4]. Note that $\zeta = 0$ when $z = 1$. Large ν approximations of the modified Bessel functions for real arguments, r, are given by

$$K_\nu(\nu r) \sim \sqrt{\frac{\pi}{2\nu}}\,\frac{e^{-\nu\phi(r)}}{\left(1+r^2\right)^{1/4}}, \qquad I_\nu(\nu r) \sim \frac{1}{\sqrt{2\pi\nu}}\,\frac{e^{\nu\phi(r)}}{\left(1+r^2\right)^{1/4}}, \qquad \nu \to \infty, \tag{3.19}$$

where

$$\phi(r) = \log\frac{r}{1+\sqrt{1+r^2}} + \sqrt{1+r^2}. \tag{3.20}$$

3.2. Hankel function logarithmic derivative. We denote the logarithmic derivative of $H_\nu^{(1)}$ by G_ν,

$$G_\nu(z) = \frac{d}{dz}\log H_\nu^{(1)}(z) = \frac{H_\nu^{(1)'}(z)}{H_\nu^{(1)}(z)}. \tag{3.21}$$

The following lemma states a few fundamental facts about G_ν that we will use below.

LEMMA 3.1. *The function $G_\nu(z)$, for $\nu \in \mathbb{R}$, satisfies the formulae*

$$G_{-\nu}(z) = G_\nu(z), \tag{3.22}$$

$$G_\nu(\overline{z}\,e^{\pi i}) = \overline{G_\nu(z)}\,e^{\pi i}, \qquad -\frac{\pi}{2} < \arg z \le \frac{\pi}{2}, \tag{3.23}$$

where $\overline{z} = |z|\,e^{-i\arg z}$ is the complex conjugate of z. Asymptotic approximations to G_ν are

$$G_\nu(z) \sim \begin{cases} \big(\log(z\,e^{-\pi i/2}/2)+\gamma\big)^{-1} z^{-1} + O(z), & \nu = 0, \\ -|\nu|\,z^{-1} + O(z^{2|\nu|-1}), & 0 < |\nu| < 1, \\ -|\nu|\,z^{-1} + O(z\log z), & |\nu| = 1, \\ -|\nu|\,z^{-1} + O(z), & |\nu| > 1, \end{cases} \qquad z \to 0, \tag{3.24}$$

where γ is the Euler constant,

$$G_\nu(z) \sim \sum_{k=0}^{\infty} i^k\frac{A_k(\nu)}{z^k}\Big(-\frac{1}{2z}+i-\frac{k}{z}\Big)\Big/\sum_{k=0}^{\infty} i^k\frac{A_k(\nu)}{z^k}, \qquad z \to \infty, \tag{3.25}$$

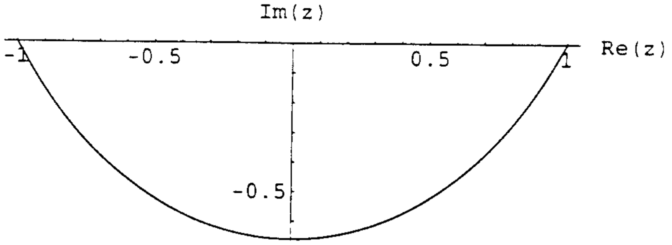


FIG. 3.1. *Curve $z(\zeta)$ defined by* (3.18) *near which the scaled zeros of $H_\nu^{(1)}$ lie (see Lemma* 3.2*). The branch cut of $H_\nu^{(1)}$ is chosen* (3.4) *on the negative imaginary axis.*

where $A_k(\nu)$ is defined in (3.7), *and*

$$G_\nu(\nu z) \sim \frac{2e^{-\pi i/3}}{\nu^{1/3}z}\left(\frac{4\zeta}{1-z^2}\right)^{-1/2}\frac{\mathrm{Ai}'(e^{2\pi i/3}\nu^{2/3}\zeta)}{\mathrm{Ai}(e^{2\pi i/3}\nu^{2/3}\zeta)}, \qquad \nu\to\infty, \tag{3.26}$$

where ζ is defined in (3.18). *Furthermore, the function u_ν defined by*

$$u_\nu(z) = z\,G_\nu(z) \tag{3.27}$$

satisfies the recurrence

$$u_\nu(z) = \frac{z^2}{\nu-1-u_{\nu-1}(z)} - \nu. \tag{3.28}$$

Proof. Equations (3.22) and (3.23) and asymptotic expansion (3.24) follow immediately from the definitions (3.2) through (3.4) of J_ν and $H_\nu^{(1)}$. The asymptotic expansion (3.25) follows from (3.5) and (3.6), while (3.26) is a consequence of (3.16) and (3.17). The recurrence (3.28) from standard Bessel recurrences [17, section 9.1.27]. □

The zeros of $H_\nu^{(1)}(z)$ are well characterized [17, 20]; they lie in the lower half z-plane near the curve shown in Figure 3.1, obtained by transformation [21] of Bessel's equation. In terms of the asymptotic approximation (3.16), this curve corresponds to negative, real arguments of the Airy function.

LEMMA 3.2. *The zeros $h_{\nu,1}, h_{\nu,2}, \dots$ of $H_\nu^{(1)}(z)$ in the sector $-\pi/2 \le \arg z \le 0$ are given by the asymptotic expansion*

$$h_{\nu,n} \sim \nu\,z(\zeta_n) + O(\nu^{-1}), \qquad \begin{array}{l}\nu\to\infty,\\ n=1,\dots,\lfloor|\nu|/2+1/4\rfloor,\end{array} \tag{3.29}$$

uniformly in n, where ζ_n is defined by the equation

$$\zeta_n = e^{-2\pi i/3}\nu^{-2/3}a_n, \tag{3.30}$$

$z(\zeta)$ is obtained from inverting (3.18), *and a_n is the nth negative zero of Airy function* Ai. *The zeros in the sector $\pi \le \arg z \le 3\pi/2$ are given by $-\overline{h_{\nu,1}}, -\overline{h_{\nu,2}}, \dots$. In particular,*

$$h_{\nu,1} \sim \nu + e^{-2\pi i/3}(\nu/2)^{1/3}(-a_1), \tag{3.31}$$

where $-a_1 = 2.338\dots$.

3.3. Pole expansions. A set of poles in a finite region defines a function that is smooth away from the region, with the smoothness increasing as the distance increases. This fact leads to the following approximation related to the fast multipole method [22, 23].

LEMMA 3.3. *Suppose that* $q_1, \dots, q_n$ *are complex numbers and* $z_1, \dots, z_n$ *are complex numbers with* $|z_j| \le 1$ *for* $j = 1, \dots, n$. *The function*

$$f(z) = \sum_{j=1}^{n} \frac{q_j}{z - z_j} \tag{3.32}$$

can be approximated for $\mathrm{Re}(z) = a > 1$ *by the* m *pole expansion*

$$g_m(z) = \sum_{j=0}^{m-1} \frac{\gamma_j}{z - \omega^j}, \tag{3.33}$$

where $\omega = e^{2\pi i/m}$ *is a root of unity and* γ_j *is defined by*

$$\gamma_j = \frac{1}{m} \sum_{l=0}^{m-1} \omega^{-jl} \sum_{k=1}^{n} q_k \, {z_k}^l, \qquad j = 0, \dots, m-1. \tag{3.34}$$

The error of the approximation is bounded by

$$|f(z) - g_m(z)| \le \frac{2(a^2+1)}{(a^m - 1)(a-1)^2} |F(z)|, \tag{3.35}$$

where

$$F(z) = \sum_{j=1}^{n} \frac{|q_j|}{z - z_j}. \tag{3.36}$$

Proof. We use the geometric series summation

$$\frac{1}{z - v} = \sum_{k=0}^{m-1} \frac{v^k}{z^{k+1}} + \frac{v^m}{z^m} \frac{1}{z - v} \tag{3.37}$$

to obtain

$$\begin{aligned} f(z) - g_m(z) = {} & \sum_{k=0}^{m-1} \frac{1}{z^{k+1}} \Big(\sum_{j=1}^{n} q_j \, {z_j}^k - \sum_{j=0}^{m-1} \gamma_j \, \omega^{jk} \Big) \\ & + \frac{1}{z^m} \Big(\sum_{j=1}^{n} \frac{q_j \, {z_j}^m}{z - z_j} - \sum_{j=0}^{m-1} \frac{\gamma_j \, \omega^{jm}}{z - \omega^j} \Big). \end{aligned} \tag{3.38}$$

All m terms of the first summation vanish, due to the combination of (3.34) and the equality $\sum_{j=0}^{m-1} \omega^{jk} = m\,\delta_{k0}$. For the error term we obtain

$$
\begin{aligned}
\left|\sum_{j=1}^{n} \frac{q_j\, {z_j}^m}{z - z_j}\right| &\le \sum_{j=1}^{n} \left|\frac{q_j\, {z_j}^m}{z - z_j}\right| \le \frac{1}{|z|} \sum_{j=1}^{n} \frac{|q_j|}{\left|1 - z_j/z\right|} \\
&\le \frac{1}{|z|} \frac{a^2+1}{(a-1)^2} \sum_{j=1}^{n} \frac{(1-a^{-1})|q_j|}{1+a^{-2}} \le \frac{a^2+1}{(a-1)^2} \frac{1}{|z|} \operatorname{Re}\left(\sum_{j=1}^{n} \frac{|q_j|}{1 - z_j/z} \right) \\
&\le \frac{a^2+1}{(a-1)^2} \left| \sum_{j=1}^{n} \frac{|q_j|}{z - z_j} \right| = \frac{a^2+1}{(a-1)^2} \, |F(z)|. \qquad (3.39)
\end{aligned}
$$

and

$$
\left| \sum_{j=0}^{m-1} \frac{\gamma_j\, \omega^{jm}}{z - \omega^j} \right| = \left| \sum_{j=0}^{m-1} \frac{\gamma_j}{z - \omega^j} \right| = \left| g_m(z) \right|. \qquad (3.40)
$$

Moreover, repeating the computations of (3.39), we find

$$
|f(z)| \le \frac{a^2+1}{(a-1)^2} |F(z)|. \qquad (3.41)
$$

Now the combination of (3.38) through (3.41) and the triangle inequality gives (3.35). □

Inequality (3.35) remains valid if we assume instead that $|z_j| \le b$ and $\operatorname{Re}(z) = ab > b$, for arbitrary $b \in \mathbb{R}$, $b > 0$; this fact leads to the next two results whose proofs mimic that of Lemma 3.3 and are omitted.

LEMMA 3.4. *Suppose* n, p *are positive integers,* $q_1, \ldots, q_n$ *are complex numbers, and* $z_1, \ldots, z_n$ *are complex numbers contained in disks* $D_1, \ldots, D_p$ *of radii* $r_1, \ldots, r_p$, *centered at* $c_1, \ldots, c_p$, *respectively. The function*

$$
f(z) = \sum_{j=1}^{n} \frac{q_j}{z - z_j} \qquad (3.42)
$$

can be approximated for z *satisfying* $\operatorname{Re}(z - c_i) \ge a r_i > r_i$ *for* $i = 1, \ldots, p$ *by the* $m \cdot p$ *pole expansion*

$$
g_m(z) = \sum_{i=1}^{p} \sum_{j=0}^{m-1} \frac{\gamma_{ij}}{z - (c_i + r_i\, \omega^j)}, \qquad (3.43)
$$

where γ_{ij} *is defined by*

$$
\gamma_{ij} = \frac{1}{m} \sum_{l=0}^{m-1} \omega^{-jl} \sum_{z_k \in D_i \setminus U_{i-1}} q_k \cdot \left(\frac{z_k - c_i}{r_i} \right)^l, \qquad \begin{array}{l} i = 1, \ldots, p, \\ j = 0, \ldots, m-1, \end{array} \qquad (3.44)
$$

with $U_i = \cup_{j \le i} D_j$. *The error of the approximation is bounded by*

$$
\left| f(z) - g_m(z) \right| \le \frac{2(a^2+1)\,|F(z)|}{(a^m - 1)(a-1)^2}, \qquad (3.45)
$$

where

$$F(z) = \sum_{j=1}^{n} \frac{|q_j|}{z - z_j}. \tag{3.46}$$

LEMMA 3.5. *Suppose that the discrete poles of Lemma* 3.4 *are replaced with a density q defined on a curve C with $C \subset U_p = D_1 \cup \cdots \cup D_p$, specifically*

$$f(z) = \int_C \frac{q(\zeta)}{z - \zeta}\, d\zeta, \tag{3.47}$$

which is finite for z outside U_p, and that g_m is defined by (3.43) *with γ_{ij} defined by*

$$\gamma_{ij} = \frac{1}{m} \sum_{l=0}^{m-1} \omega^{-jl} \int_{C \cap (D_i \setminus U_{i-1})} q(\zeta) \left(\frac{\zeta - c_i}{r_i} \right)^l d\zeta, \qquad \begin{array}{l} i = 1, \ldots, p, \\ j = 0, \ldots, m-1, \end{array} \tag{3.48}$$

with $U_i = \cup_{j \le i} D_j$. Then the bound (3.45) *holds as before.* Lemma 3.3 enables us to approximate, with exponential convergence, a function defined as a sum of poles. The fundamental assumption is that the region of interest be "separated" from the pole locations. The notion of separation is effectively relaxed by covering the pole locations with disks of varying size in an adaptive manner. In Lemmas 3.4 and 3.5, we use this approach to derive our principal analytical result.

4. Rational approximation of the logarithmic derivative.

The Hankel function's logarithmic derivative $G_\nu(z)$ defined in (3.21) approaches a constant as $z \to \infty$ and is regular for finite $z \in \mathbb{C}$, except at $z = 0$, which is a branch point, and at the zeros of $H_\nu^{(1)}(z)$, all of which are simple. We can therefore develop a representation for G_ν analogous to that of the Mittag–Leffler theorem; the only addition is due to the branch cut on the negative imaginary axis. It will be convenient to work with $u_\nu(z)$, defined in (3.27), for which approximations to be introduced have simple error bounds.

THEOREM 4.1. *The function $u_\nu(z) = z\, G_\nu(z)$, where G_ν is defined for $\nu \in \mathbb{R}$ by* (3.21) *with the branch cut defined by* (3.4), *is given by the formula*

$$u_\nu(z) = iz - \frac{1}{2} + \sum_{n=1}^{N_\nu} \frac{h_{\nu,n}}{z - h_{\nu,n}} - \frac{1}{\pi i} \int_0^\infty \frac{\operatorname{Im}\big(u_\nu(re^{-\pi i/2})\big)}{ir + z}\, dr \tag{4.1}$$

for $z \in \mathbb{C}$ not in $\{0, h_{\nu,1}, h_{\nu,2}, \ldots, h_{\nu,N_\nu}\}$ and not on the negative imaginary axis. Here $h_{\nu,1}, h_{\nu,2}, \ldots, h_{\nu,N_\nu}$ denote the zeros of $H_\nu^{(1)}(z)$, which number N_ν.

Proof. The case of the spherical Hankel function, where $\nu = k + 1/2$ for $k \in \mathbb{Z}$, is simple and we consider it first. Here $u_\nu(z)$ is a ratio of polynomials in iz with real coefficients, which is clear from the observation that $u_{1/2}(z) = iz - 1/2$ in combination with the recurrence (3.28). Hence

$$u_\nu(z) = p(z) + \sum_{n=1}^{N_\nu} \frac{\alpha_{\nu,n}}{z - h_{\nu,n}}, \tag{4.2}$$

where p is a polynomial and $\alpha_{\nu,n}$ is the residue of u_ν at $h_{\nu,n}$,

$$\alpha_{\nu,n} = \lim_{z \to h_{\nu,n}} (z - h_{\nu,n})\, u_\nu(z) = h_{\nu,n} \tag{4.3}$$

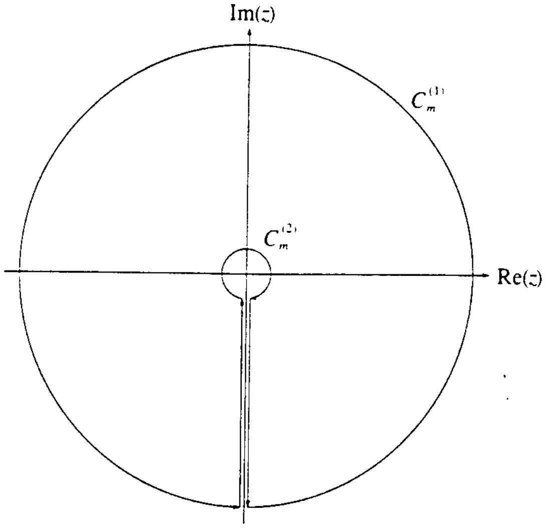


FIG. 4.1. *Integration contour C_m, with inner circle radius $1/m$ and outer radius $m+1$.*

by l'Hôpital's rule. We see from (3.25) that

$$u_\nu(z) \sim iz - \frac{1}{2} + O(z^{-1}), \qquad z \to \infty, \tag{4.4}$$

whence

$$p(z) = iz - \frac{1}{2}. \tag{4.5}$$

Noting that $u_\nu(iy) \in \mathbb{R}$ for $y \in \mathbb{R}$, and combining (4.2), (4.3), and (4.5), we obtain (4.1).

We now consider the case $\nu \neq k+1/2$, $k \in \mathbb{Z}$, for which the origin is a branch point. For $m = 1, 2, \ldots$, we define C_m to be the simple closed curve, shown in Figure 4.1, which proceeds counterclockwise along the circle $C_m^{(1)}$ of radius $m+1$ centered at the origin from $\arg z = -\pi/2$ to $3\pi/2$, to the vertical segment $z = re^{3\pi i/2}$, $r \in [1/m, m+1]$, to the circle $C_m^{(2)}$ of radius $1/m$ centered at the origin from $\arg z = 3\pi/2$ to $-\pi/2$, to the vertical segment $z = re^{-\pi i/2}$, back to the first circle. Since none of the zeros of $H_\nu^{(1)}$ lies on the imaginary axis, C_m encloses them all if m is sufficiently large. For such m, and $z \in \mathbb{C}$ inside C_m with $H_\nu^{(1)}(z) \neq 0$, the residue theorem gives

$$\frac{1}{2\pi i} \int_{C_m} \frac{u_\nu(\zeta)}{\zeta - z}\, d\zeta = u_\nu(z) + \sum_{n=1}^{N_\nu} \frac{h_{\nu,n}}{h_{\nu,n} - z}. \tag{4.6}$$

We now consider the separate pieces of the contour C_m. For the circles $C_m^{(1)}$ and $C_m^{(2)}$, we use the asymptotic expansion (4.4) about infinity and (3.24) about the origin to obtain

$$\lim_{m\to\infty} \frac{1}{2\pi i} \int_{C_m^{(1)}} \frac{u_\nu(\zeta)}{\zeta - z}\, d\zeta = iz - \frac{1}{2}, \qquad \lim_{m\to\infty} \frac{1}{2\pi i} \int_{C_m^{(2)}} \frac{u_\nu(\zeta)}{\zeta - z}\, d\zeta = 0. \tag{4.7}$$

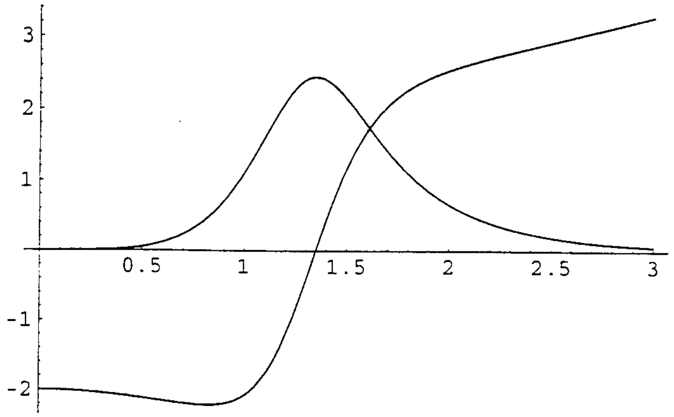


FIG. 4.2. *Plot of* $\mathrm{Re}\big(u_\nu(re^{-\pi i/2})\big)$, *containing the zero crossing, and* $\mathrm{Im}\big(u_\nu(re^{-\pi i/2})\big)$, *for* $\nu = 2$ *and* $r \in [0, 3]$.

Now exploiting the symmetry $u_\nu(re^{3\pi i/2}) = \overline{u_\nu(re^{-\pi i/2})}$ from (3.23) for the vertical segments, we obtain

$$\lim_{m\to\infty} \frac{1}{2\pi i} \int_{C_m} \frac{u_\nu(\zeta)}{\zeta - z}\, d\zeta = iz - \frac{1}{2} + \frac{1}{2\pi i} \int_0^\infty \frac{2i\,\mathrm{Im}\big(u_\nu(re^{-\pi i/2})\big)}{(re^{-\pi i/2} - z)}\, e^{-\pi i/2}\, dr, \tag{4.8}$$

which, when combined with (4.6), yields (4.1) and the theorem. □

The primary aim of this paper is to reduce the summation and integral of (4.1) to a similar summation involving dramatically fewer terms. To do so, we restrict z to the upper half-plane and settle for an approximation. Such a representation is possible, for the poles of u_ν (zeros of $H_\nu^{(1)}$) lie entirely in the lower half-plane and do not cluster near the real axis. We first examine the behavior of u_ν on the negative imaginary axis.

The qualitative behavior of u_ν on the branch cut is illustrated by the case of $\nu = 2$, shown in Figure 4.2. The plot changes little with changing ν, except for the sign of $\mathrm{Im}(u_\nu(z))$ and the sharpness of its extremum.

LEMMA 4.2. *For* $\nu \in \mathbb{R}$, $\nu \neq k + 1/2$, $k \in \mathbb{Z}$, *the function* $u_\nu(re^{-\pi i/2})$ *is infinitely differentiable on* $r \in (0, \infty)$ *and has imaginary part satisfying the following formulae:*

$$\mathrm{Im}\big(u_\nu(re^{-\pi i/2})\big) = \frac{\pi\cos(\nu\pi)}{\cos^2(\nu\pi)K_\nu^2(r) + \big(\pi I_\nu(r) + \sin(\nu\pi)K_\nu(r)\big)^2} \neq 0, \tag{4.9}$$

$$\mathrm{Im}\big(u_\nu(re^{-\pi i/2})\big) \sim \begin{cases} \dfrac{\pi}{(\log(r/2) + \gamma)^2 + \pi^2}, & \nu = 0, \\ \dfrac{\pi\cos(\nu\pi)}{4^{|\nu|-1}\Gamma(|\nu|)^2}\, r^{2|\nu|}, & \nu \neq 0, \end{cases} \qquad r \to 0, \tag{4.10}$$

$$\mathrm{Im}\big(u_\nu(re^{-\pi i/2})\big) \sim 2\cos(\nu\pi)\, re^{-2r}, \qquad r \to \infty, \tag{4.11}$$

$$\mathrm{Im}\big(u_\nu(re^{-\pi i/2})\big) \sim \frac{\cos(\nu\pi)\sqrt{r^2 + \nu^2}}{\cosh\big(2\nu\,\phi(r/|\nu|)\big) + \sin(|\nu|\pi)}, \qquad |\nu| \to \infty, \tag{4.12}$$

where ϕ *is defined in* (3.20).

Proof. Infinite differentiability of $u_\nu(z)$ follows from the observation that $H_\nu^{(1)}(z) \neq 0$ on the negative imaginary axis. To derive (4.9) we recall (3.11) to obtain

$$\operatorname{Im}(u_\nu(re^{-\pi i/2})) = \frac{r\pi\cos(\nu\pi)(K_\nu(r)I'_\nu(r) - K'_\nu(r)I_\nu(r))}{\cos^2(\nu\pi)K_\nu^2(r) + (\pi I_\nu(r) + \sin(\nu\pi)K_\nu(r))^2}, \tag{4.13}$$

then apply (3.10). The remaining formulas follow from the asymptotic forms of $K_\nu(r)$ and $I_\nu(r)$ for small and large r, and the uniform large ν expansions given in (3.12) through (3.15) and (3.19). Here we use the symmetry $u_{-\nu} = u_\nu$. Note that (4.10) is valid for $r/|\nu| \to 0$. The approximation (4.12) is nonuniform for $\nu \approx 2k - 1/2$ and $\pi I_\nu(r) + \sin(\nu\pi)K_\nu(r) \approx 0$. □

LEMMA 4.3. *Given $\nu_0 > 0$ there exist constants c_0 and c_1 such that for all $\nu \in \mathbb{R}$, $|\nu| \geq \nu_0$, $\nu \neq k + 1/2$, $k \in \mathbb{Z}$, and all z satisfying* $\operatorname{Im}(z) \geq 0$, *the function*

$$f(z) = \int_0^\infty \frac{\operatorname{Im}(u_\nu(re^{-\pi i/2}))}{ir + z}\,dr \tag{4.14}$$

satisfies the bounds

$$\frac{c_0}{1 + |z|/|\nu|} \leq |f(z)| \leq \frac{c_1}{1 + |z|/|\nu|}. \tag{4.15}$$

Moreover, there exists $\delta > 0$ such that for all $\nu \in \mathbb{R}$, $|\nu| \geq \nu_0$, and ε with $0 < \varepsilon < 1/2$, $f(z)$ admits an approximation $g(z)$ that is a sum of $d \leq \delta \cdot (1 + |\nu|^{-1}\log(1/\varepsilon)) \cdot \log(1/\varepsilon)$ poles, with

$$|f(z) - g(z)| \leq \varepsilon \cdot |f(z)|, \tag{4.16}$$

provided $\operatorname{Im}(z) \geq 0$.

Proof. We assume $\nu \neq k + 1/2$ for integral k and begin by changing variables, $r = |\nu|w$, so that

$$f(z) = \int_0^\infty \frac{\operatorname{Im}(u_\nu(|\nu|we^{-\pi i/2}))}{iw + z/|\nu|}\,dw = \int_0^\infty \mu_z(w)dw. \tag{4.17}$$

From the nonvanishing of μ_z and its asymptotic behavior in w, it is clear that (4.15) holds for $|\nu| \in (\nu_0, \nu_1)$ and any fixed $\nu_1 > \nu_0$. Using (4.12) for $|\nu|$ large but bounded away from $2k - 1/2$ for integral k, an application of Watson's lemma to (4.14) focuses on the unique positive zero, w^*, of ϕ defined in (3.20). As the derivative of this function is positive, we conclude

$$f(z) \sim \frac{\alpha\cos(\nu\pi)}{iw^* + z/|\nu|}, \tag{4.18}$$

where α is a function of w^*, so that (4.15) clearly holds. However, as $\nu \to 2k - 1/2$, the denominator on the right-hand side of (4.12) may nearly vanish at w^* and the expansion loses its uniformity. Setting $\cos(\nu\pi) = \eta$ in these cases, we see that the denominator has a minimum which is bounded below by $O(\eta^2)$. Hence in an $O(|\nu|^{-1})$ neighborhood of the minimum which includes w^*, we have

$$\int \mu_z(w) \approx \frac{\eta|\nu|\sqrt{1 + (w^*)^2}}{iw^* + z/|\nu|} \int_{-\gamma/|\nu|}^{\gamma/|\nu|} \frac{1}{\eta^2 + \beta^2\nu^2 s^2}\,ds, \tag{4.19}$$

which by the change of variables $s = \eta z/|\nu|$ is seen to satisfy the upper bound in (4.15) uniformly in η. As the rest of the integral is small, the upper bound holds.

We now move on to the approximation. For a positive integer m and a positive number w_0, we define intervals $I_0 = (0, w_0)$, $I_j = (2^{j-1}w_0, 2^j w_0)$ for $j = 1, \dots, m$, and $I_{m+1} = (2^m w_0, \infty)$. Now

$$f(z) = f_0(z) + f_1(z) + f_2(z), \tag{4.20}$$

where f_0, f_1, and f_2 are defined by the formulae

(4.21)

$$f_0(z) = \int_{I_0} \mu_z(w)\, dw, \qquad f_1(z) = \sum_{j=1}^{m} \int_{I_j} \mu_z(w)\, dw, \qquad f_2(z) = \int_{I_{m+1}} \mu_z(w)\, dw.$$

We will now choose w_0 and m so that f_0 and f_2 can be ignored and then use Lemma 3.5 to approximate f_1. Using (4.10) and (4.12) and taking w_0 sufficiently small we have, for some constant c_2 independent of ν,

$$|f_0(z)| \le \frac{c_2|\nu|}{1+|z|/|\nu|}\left(\frac{3e}{4}\right)^{2|\nu|} \int_0^{w_0} w^{2|\nu|-1} dw \le \frac{c_2}{1+|z|/|\nu|}\left(\frac{3e}{4}w_0\right)^{2|\nu|}. \tag{4.22}$$

Hence, a choice of

$$w_0 = O(\varepsilon^{1/(2|\nu|)}), \quad \varepsilon \to 0, \tag{4.23}$$

suffices to guarantee

$$|f_0(z)| \le \frac{\varepsilon}{3}|f(z)| \tag{4.24}$$

in the closed upper half-plane. Now using (4.11) and (4.12) and assuming m sufficiently large we have, for some constant c_3 independent of ν,

$$|f_2(z)| \le \frac{c_3|\nu|}{1+|z|/|\nu|} \int_{2^m w_0}^{\infty} w e^{-|\nu| w} dw \le \frac{c_2}{1+|z|/|\nu|} 2^m w_0 e^{-|\nu| 2^m w_0}. \tag{4.25}$$

From (4.23), choosing

$$m \ge m_0 + m_1 \frac{1}{|\nu|} \log \frac{1}{\varepsilon} \tag{4.26}$$

for appropriate m_0 and m_1 independent of ν and ε leads to

$$|f_2(z)| \le \frac{\varepsilon}{3}|f(z)|. \tag{4.27}$$

Finally, we apply Lemma 3.5 to the approximation of f_1. The error involves the function $F_1 = \int |\mathrm{Im}(u_\nu)|/(ir + z) dr$, but we note that $|F_1| = |f_1|$. Using p poles for each j we produce a $p \cdot m$-pole approximation $g(z)$ with an error estimate, again for $\mathrm{Im}(z) \ge 0$, given by

$$|f_1(z) - g(z)| \le \frac{5}{3^p - 1}|f_1(z)|. \tag{4.28}$$

A choice of

$$p = O\Big(\log\frac{1}{\varepsilon}\Big) \tag{4.29}$$

enforces

$$|f_1(z) - g(z)| \le \frac{\varepsilon}{3}|f(z)|. \tag{4.30}$$

By combining (4.24), (4.27), (4.30), and the triangle inequality, we obtain (4.16) with the number of poles, $d = p \cdot m$, satisfying the stated bound. □

The case $\nu = 0$ requires special treatment. First, the direct application of the preceding arguments leads to a significantly larger upper bound on the number of poles. Second, we note that $u_0(0) = 0$, so that relative error bounds near $z = 0$ require a vanishing absolute error. Finally, the lack of regularity of $u_0(z)$ at $z = 0$ precludes uniform rational approximation, as discussed in [10]. Therefore, we relax the condition $\mathrm{Im}(z) \ge 0$ to $\mathrm{Im}(z) \ge \eta > 0$. By (2.20) this will lead to good approximate convolutions for times $T \le \eta^{-1}$.

LEMMA 4.4. *There exists $\delta > 0$ such that for all ε, $0 < \varepsilon < 1/2$ and η, $0 < \eta < 1/2$, the function $f(z) = u_0(z) - iz + 1/2$ admits an approximation $g(z)$ that is a sum of $d \le \delta \cdot \big(\log(1/\eta) + \log\log(1/\varepsilon)\big) \cdot \log(1/\varepsilon)$ poles, with*

$$|f(z) - g(z)| \le \varepsilon \cdot |f(z)|, \tag{4.31}$$

provided $\mathrm{Im}(z) \ge \eta$.

Proof. Note that since $u_0(z)$ has no poles, $f(z)$ is given by (4.14) and satisfies (4.15). Define intervals

$$I_j = \big((2^{j-1} - 1)\eta, (2^j - 1)\eta\big) \text{ for } j = 1, \dots, m, \quad I_{m+1} = \big((2^m - 1)\eta, \infty\big).$$

Now

$$f(z) = f_1(z) + f_2(z), \tag{4.32}$$

where f_1 and f_2 are defined by the formulae

$$f_1(z) = \sum_{j=1}^{m} \int_{I_j} \frac{\mathrm{Im}(u_0(re^{-\pi i/2}))}{ir + z}\, dr, \qquad f_2(z) = \int_{I_{m+1}} \frac{\mathrm{Im}(u_0(re^{-\pi i/2}))}{ir + z}\, dr. \tag{4.33}$$

We will now choose m so that f_2 can be ignored and then use Lemma 3.5 to approximate f_1. Using (4.11) and assuming m sufficiently large we have, for some constant c,

$$|f_2(z)| \le \frac{c}{1 + |z|} \int_{(2^m - 1)\eta}^{\infty} re^{-2r} dw \le \frac{c}{1 + |z|} 2^{m-1} \eta e^{-2^m \eta}. \tag{4.34}$$

Hence, choosing

$$m \ge m_0\big(\log(1/\eta) + \log\log(1/\varepsilon)\big) \tag{4.35}$$

for appropriate m_0 independent of η and ε leads to

$$|f_2(z)| \le \frac{\varepsilon}{2}|f(z)|. \tag{4.36}$$

Finally, we apply Lemma 3.5 to the approximation of f_1. Using p poles for each j we produce a $p \cdot m$-pole approximation $g(z)$ with an error estimate for $\mathrm{Im}(z) \geq \eta$ given by

$$|f_1(z) - g(z)| \leq \frac{5}{3^p - 1}|f_1(z)|. \tag{4.37}$$

A choice of

$$p = O\Big(\log \frac{1}{\varepsilon}\Big) \tag{4.38}$$

enforces

$$|f_1(z) - g(z)| \leq \frac{\varepsilon}{2}|f(z)|. \tag{4.39}$$

By (4.36), (4.39), and the triangle inequality, (4.31) is achieved with the number of poles, $d = p \cdot m$, satisfying the stated bound. ☐

We now consider the contribution of the poles.

LEMMA 4.5. *There exist constants* C_0, C_1, $\delta > 0$ *such that for all* $\nu, \varepsilon \in \mathbb{R}$ *with* $2 \leq |\nu|$ *and* $0 < \varepsilon < 1/2$ *the function*

$$h(z) = \sum_{n=1}^{N_\nu} \frac{h_{\nu,n}}{z - h_{\nu,n}}, \tag{4.40}$$

where $h_{\nu,1}, \ldots, h_{\nu,N_\nu}$ *are the roots of* $H_\nu^{(1)}$, *satisfies the inequalities*

$$\frac{C_1|\nu|}{1 + |z|/|\nu|} \leq |h(z)| \leq \frac{C_2|\nu|}{1 + |z|/|\nu|}, \tag{4.41}$$

and admits an approximation $g(z)$ *that is a sum of* $d \leq \delta \cdot \log|\nu| \cdot \log(1/\varepsilon)$ *poles, with*

$$|h(z) - g(z)| \leq \varepsilon \cdot |h(z)|, \tag{4.42}$$

provided $\mathrm{Im}(z) \geq 0$.

Proof. The curve C defined in Lemma 3.2, near which $h_{\nu,1}/|\nu|, \ldots, h_{\nu,N_\nu}/|\nu|$ lie, is contained in disks separated from the real axis. If we denote the disk of radius r centered at c by $D(r, c)$, then the disks

$$\{D(-\mathrm{Im}(z), z) |\; z \in C, |\arg z - \pi/2| = \pi/2 + \pi/2^n,\ n = 1, 2, \ldots\}, \tag{4.43}$$

for example, contain $C \backslash \{+1, -1\}$. From (3.31), the root $h_{\nu,1}$ closest to the real axis satisfies

$$\arg h_{\nu,1} \sim \frac{a_1\sqrt{3}}{2^{4/3}}|\nu|^{-2/3}; \tag{4.44}$$

hence it is contained in a disk of (4.43) with $n \approx \log_2\left(2^{4/3}3^{-1/2}\pi(-a_1)^{-1}|\nu|^{2/3}\right)$, and all of the roots are contained in $O(\log|\nu|)$ of the disks. Now applying Lemma 3.4 we obtain (4.42) with $|h|$ replaced by $|H| = \left|\sum |h_{\nu,n}|/(z - h_{\nu,n})\right|$. To obtain the upper bound in (4.41) for both h and H we note first that it is trivial except for $|z/\nu| \approx 1$. A detailed analysis of the roots as described by Lemma 3.2 shows that

$$|\mathrm{Im}(h_{\nu,j})| \geq cj^{2/3}|\nu|^{1/3}. \tag{4.45}$$

Hence, for $|z/\nu| \approx 1$,

$$\sum_j \left| \frac{h_{\nu,j}}{z - h_{\nu,j}} \right| \leq C|\nu|^{2/3} \sum_{j=1}^{|\nu|} j^{-2/3} \leq 3C|\nu|. \tag{4.46}$$

The lower bound in (4.41) is again obvious except for $|z/\nu| \approx 1$. Then, however, we note that

$$h(z) = u_\nu(z) - iz + 1/2 - f(z). \tag{4.47}$$

Since, from (3.26), $|u_\nu(z)| = O(|\nu|^{2/3})$ for $|z/\nu| \approx 1$ and $|f(z)| = O(1)$ by (4.15) the right-hand side is dominated by $-iz$ and $|h(z)| = O(|\nu|)$. □

The combination of Theorem 4.1 and Lemmas 4.3 and 4.5 suffices to prove our principal analytical result.

THEOREM 4.6. *Given* $\nu_0 > 0$ *there exists* $\delta > 0$ *such that for all* $\nu \in \mathbb{R}$, $|\nu| \geq \nu_0$, *and* $0 < \varepsilon < 1/2$ *there exists* d *with*

$$d \leq \delta\big(\log|\nu| \cdot \log(1/\varepsilon) + \log^2|\nu| + |\nu|^{-1}\log^2(1/\varepsilon)\big), \tag{4.48}$$

and complex numbers $\alpha_1, \ldots, \alpha_d$ *and* $\beta_1, \ldots, \beta_d$, *depending on* ν *and* ε, *such that the function*

$$U_{\nu,\varepsilon}(z) = iz - \frac{1}{2} + \sum_{n=1}^{d} \frac{\alpha_n}{z - \beta_n} \tag{4.49}$$

approximates $u_\nu(z)$ *with the bound*

$$\big|u_\nu(z) - U_{\nu,\varepsilon}(z)\big| \leq \varepsilon \cdot \big|u_\nu(z)\big|, \tag{4.50}$$

provided that $\mathrm{Im}(z) \geq 0$. *Furthermore*

$$\left(\int_{-\infty}^{\infty} \big|u_\nu(x) - U_{\nu,\varepsilon}(x)\big|^2 dx\right)^{1/2} \leq \varepsilon \cdot \left(\int_{-\infty}^{\infty} \big|u_\nu(x) - ix + 1/2\big|^2 dx\right)^{1/2}. \tag{4.51}$$

Proof. We first note the lower bound

$$\big|u_\nu(z) - iz + 1/2\big| \geq \frac{c|\nu|}{1 + |z|/|\nu|}. \tag{4.52}$$

For $\nu > 0$ the function is nonvanishing and has the correct asymptotic behavior, so we need only consider the case of $|\nu|$ large. The result then follows from (3.26). This proves (4.51) and (4.50) with u_ν replaced by $u_\nu - iz + 1/2$ on the right-hand side. From (3.26) we have

$$\big|u_\nu(z) - iz + 1/2\big| \leq c|\nu|^{1/3}\big|u_\nu(z)\big|, \tag{4.53}$$

so that the final result follows from the scaling $\varepsilon \to |\nu|^{-1/3}\varepsilon$. □

The number of poles in (4.48) required to approximate $u_\nu(z)$ to a tolerance ε depends on both ε and ν. The asymptotic dependence on ε is proportional to $|\nu|^{-1}\log^2(1/\varepsilon)$. We will see in the numerical examples, however, that this term is important only for small $|\nu|$; otherwise the dominant term is the first, for an asymptotic dependence of $O(\log|\nu|\cdot\log(1/\varepsilon))$. As we generally have $\varepsilon \ll |\nu|^{-1}$ in practice, the term $\log^2|\nu|$ is of less importance.

Similarly, Lemma 4.4 leads to the following theorem for $\nu = 0$.

THEOREM 4.7. *There exists $\delta > 0$ such that for all ε, $0 < \varepsilon < 1/2$ and η, $0 < \eta < 1/2$ there exists $d \leq \delta \cdot (\log(1/\eta)\cdot\log(1/\varepsilon) + \log\log(1/\varepsilon) + \log\log(1/\eta))$ and complex numbers $\alpha_1, \ldots, \alpha_d$ and $\beta_1, \ldots, \beta_d$, depending on η and ε, such that the function*

$$U_{0,\varepsilon}(z) = iz - \frac{1}{2} + \sum_{n=1}^{d} \frac{\alpha_n}{z-\beta_n} \tag{4.54}$$

approximates $u_0(z)$ with the bound

$$\left|u_0(z) - U_{0,\varepsilon}(z)\right| \leq \varepsilon \cdot \left|u_0(z)\right|, \tag{4.55}$$

provided that $\mathrm{Im}(z) \geq \eta$. *Furthermore*

$$\begin{aligned}\left(\int_{-\infty}^{\infty}\left|u_0(x+i\eta) - U_{0,\varepsilon}(x+i\eta)\right|^2 dx\right)^{1/2} \\ \leq \varepsilon \cdot \left(\int_{-\infty}^{\infty}\left|u_\nu(x+i\eta) - ix + \eta + 1/2\right|^2 dx\right)^{1/2}.\end{aligned} \tag{4.56}$$

Proof. Again we already have (4.55) with $u_0(z) - iz + 1/2$ on the right-hand side. By (3.24) we find

$$\left|u_0(z) - iz + 1/2\right| \leq c\log(1/\eta)\left|u_0(z)\right|. \tag{4.57}$$

The theorem follows from the scaling $\varepsilon \to \log^{-1}(1/\eta)\varepsilon$. ☐

As we must take $\eta = T^{-1}$, we see that the number of poles required may grow like $\log(1/\varepsilon)\cdot\log T + \log T\cdot\log\log T$. However, this is only for the mode $n = 0$ in the two-dimensionsal case. In short, the T dependence is insignificant in practice.

5. Computation of the rational representations. Analytical error bound estimates developed in the previous sections are based on maximum norm errors as in (2.19) and (2.20). In numerical computation it is often convenient, however, to obtain least squares solutions. Our method of computing a rational function $U_{\nu,\varepsilon}$ that satisfies (4.50) is to enforce (4.51). An alternative approach would be to use rational Chebyshev approximation as developed by Trefethen and Gutknecht [24, 25, 26].

In the numerical computations, we work with

$$\tilde{u}_\nu(z) = u_\nu(z) - iz + 1/2 \tag{5.1}$$

and its sum-of-poles approximation $\tilde{U}_{\nu,\varepsilon}(z) = U_{\nu,\varepsilon}(z) - iz + 1/2$. In particular, we have the nonlinear least squares problem

$$\min_{P,Q}\int_{-\infty}^{\infty}\left|\frac{P(x)}{Q(x)} - \tilde{u}_\nu(x)\right|^2 dx \tag{5.2}$$

for P, Q polynomials with $\deg(P) + 1 = \deg(Q) = d$. Problem (5.2) is not only nonlinear, but also very poorly conditioned when P, Q are represented in terms of their monomial coefficients. We apply two tactics for coping with these difficulties: linearization and orthogonalization.

We linearize the problem by starting with a good estimate of Q and updating P, Q iteratively. In particular, we solve the linear least squares problem

$$\min_{P^{(i+1)}, Q^{(i+1)}} \int_{-\infty}^{\infty} \left| \frac{P^{(i+1)}(x)}{Q^{(i)}(x)} - \frac{Q^{(i+1)}(x)}{Q^{(i)}(x)} \, \tilde{u}_\nu(x) \right|^2 dx, \tag{5.3}$$

where the integral is replaced by a quadrature. The initial values $P^{(0)}$, $Q^{(0)}$ are obtained by exploiting the asymptotic expansion (3.25) and the recurrence (3.28). We find that two to three iterations of (5.3) generally suffice.

The quadrature for (5.3) is derived by first changing variables,

$$\int_{-\infty}^{\infty} f(x)\, dx = \int_{-\pi/2}^{\pi/2} f(\tan\theta)\, \sec^2\theta \, d\theta \approx \sum_{i=1}^{m} w_i \, f(\tan\theta_i)\, \sec^2\theta_i, \tag{5.4}$$

where $\theta_1, \ldots, \theta_m$ and $w_1 \ldots, w_m$ denote appropriate quadrature nodes and weights. The transformed integrand is periodic on the interval $[-\pi/2, \pi/2]$, so the trapezoidal rule (or midpoint rule) is an obvious candidate. The integrand is infinitely continously differentiable, except at $\theta = 0$, where its regularity is of order $2|\nu|$. For $|\nu| > 8$ (say), the trapezoidal rule delivers at least 16th-order convergence and is very effective. For small $|\nu|$, however, a quadrature that adjusts for the complicated singularity at $\theta = 0$ is needed. Here we can successively subdivide the interval near the singularity, applying high-order quadratures on each subinterval (see, for example, [27]).

The quadrature discretization of (5.3) cannot be solved as a least squares problem by standard techniques, due to its extremely poor conditioning. We avoid forming the corresponding matrix; rather we solve the least squares problem by Gram-Schmidt orthogonalization. The $2d+1$ functions

$$\tilde{u}_\nu,\ 1,\ x\tilde{u}_\nu,\ x, \ldots, x^{d-1}\tilde{u}_\nu,\ x^{d-1},\ x^d\tilde{u}_\nu \tag{5.5}$$

are orthogonalized under the real inner product

$$\langle f, g\rangle_i = \int_{-\infty}^{\infty} \frac{\operatorname{Re}\big(f(x)\,\bar{g}(x)\big)}{\left|Q^{(i)}(x)\right|^2}\, dx \tag{5.6}$$

to obtain the orthogonal functions

$$g_n(x) = \begin{cases} \tilde{u}_\nu(x), & n = 1, \\ 1, & n = 2, \\ x g_{n-2}(x) - \sum_{j=1}^{\min\{4,n-1\}} c_{nj}\, g_{n-j}(x), & n = 3, \ldots, 2d+1, \end{cases} \tag{5.7}$$

where

$$c_{nj} = \frac{\langle x g_{n-2}, g_{n-j}\rangle_i}{\langle g_{n-j}, g_{n-j}\rangle_i}, \qquad \begin{array}{l} n = 3, \ldots, 2d+1, \\ j = 1, \ldots, \min\{4, n-1\}. \end{array} \tag{5.8}$$

Now

$$g_{2d+1} = -P^{(i+1)} + \tilde{u}_\nu\, Q^{(i+1)}, \tag{5.9}$$

TABLE 1

Number d of poles to represent the Laplace transform of nonreflecting boundary kernels σ_n and ω_n, for various values of ε.

$\varepsilon = 10^{-6}$			
σ_n		ω_n	
n	d	n	d
0	26		
1	9		
2	6		
3– 6	5	0– 5	n
7– 8	6	6– 8	6
9– 12	7	9– 12	7
13– 19	8	13– 19	8
20– 31	9	20– 31	9
32– 51	10	32– 51	10
52– 86	11	52– 86	11
87– 147	12	87– 147	12
148– 227	13	148– 228	13
228– 401	14	229– 402	14
402– 728	15	403– 728	15
729–1024	16	729–1024	16

$\varepsilon = 10^{-8}$			
σ_n		ω_n	
n	d	n	d
0	44		
1	15		
2	9		
3– 8	7	0– 7	n
9– 10	8	8– 10	8
11– 14	9	11– 14	9
15– 20	10	15– 19	10
21– 28	11	20– 28	11
29– 41	12	29– 40	12
42– 58	13	41– 57	13
59– 84	14	58– 83	14
85– 123	15	84– 123	15
124– 183	16	124– 183	16
184– 275	17	184– 275	17
276– 418	18	276– 418	18
419– 638	19	419– 637	19
639– 971	20	638– 971	20
972–1024	21	972–1024	21

$\varepsilon = 10^{-15}$			
σ_n		ω_n	
n	d	n	d
1	41		
2	24		
3	18		
4	15		
5	14		
6	13		
7– 12	12		
13– 14	13	0– 13	n
15– 16	14	14– 15	14
17– 18	15	16– 18	15
19– 22	16	19– 21	16
23– 26	17	22– 25	17
27– 31	18	26– 30	18
32– 37	19	31– 36	19
38– 45	20	37– 44	20
46– 54	21	45– 53	21
55– 65	22	54– 65	22
66– 79	23	66– 79	23
80– 97	24	80– 96	24
98– 118	25	97– 118	25
119– 145	26	119– 144	26
146– 177	27	145– 176	27
178– 216	28	177– 216	28
217– 265	29	217– 264	29
266– 324	30	265– 324	30
325– 397	31	325– 396	31
398– 486	32	397– 485	32
487– 595	33	486– 594	33
596– 728	34	595– 727	34
729– 890	35	728– 890	35
891–1024	36	891–1024	36

so $P^{(i+1)}$ and $Q^{(i+1)}$ are computed from the recurrence coefficients c_{nj} by splitting (5.7) into even- and odd-numbered parts.

For some applications, including nonreflecting boundary kernels, it is convenient to represent P/Q as a sum of poles,

$$\frac{P(z)}{Q(z)} = \sum_{n=1}^{d} \frac{\alpha_n}{z - \beta_n}. \tag{5.10}$$

We compute $\beta_1, \ldots, \beta_d$ (zeros of Q) by Newton iteration with zero suppression (see,

TABLE 2

Laplace transform of cylinder kernel σ_n defined in (2.13), approximated as a sum of d poles, for $n = 1, \dots, 4$ and $\varepsilon = 10^{-6}$.

		Pole Coefficient		Pole Location	
n	d	Re	Im	Re	Im
1	9	$-0.426478E-02$	$0.000000E+00$	$-0.368403E+01$	$0.000000E+00$
		$-0.416255E-01$	$0.000000E+00$	$-0.205860E+01$	$0.000000E+00$
		$-0.122665E+00$	$0.000000E+00$	$-0.118994E+01$	$0.000000E+00$
		$-0.143704E+00$	$0.000000E+00$	$-0.717570E+00$	$0.000000E+00$
		$-0.530662E-01$	$0.000000E+00$	$-0.423506E+00$	$0.000000E+00$
		$-0.863872E-02$	$0.000000E+00$	$-0.223111E+00$	$0.000000E+00$
		$-0.961472E-03$	$0.000000E+00$	$-0.103710E+00$	$0.000000E+00$
		$-0.721548E-04$	$0.000000E+00$	$-0.409342E-01$	$0.000000E+00$
		$-0.250102E-05$	$0.000000E+00$	$-0.117156E-01$	$0.000000E+00$
2	6	$0.218164E-01$	$0.000000E+00$	$-0.333263E+01$	$0.000000E+00$
		$0.860648E+00$	$0.000000E+00$	$-0.162945E+01$	$0.000000E+00$
		$-0.138934E+01$	$0.162069E+00$	$-0.125843E+01$	$0.412637E+00$
		$-0.138934E+01$	$-0.162069E+00$	$-0.125843E+01$	$-0.412637E+00$
		$0.209905E-01$	$0.000000E+00$	$-0.612710E+00$	$0.000000E+00$
		$0.232032E-03$	$0.000000E+00$	$-0.240327E+00$	$0.000000E+00$
3	5	$-0.179277E+00$	$0.000000E+00$	$-0.309775E+01$	$0.000000E+00$
		$-0.168335E+01$	$0.129111E+01$	$-0.167998E+01$	$0.130784E+01$
		$-0.168335E+01$	$-0.129111E+01$	$-0.167998E+01$	$-0.130784E+01$
		$-0.816322E+00$	$0.000000E+00$	$-0.187260E+01$	$0.000000E+00$
		$-0.126962E-01$	$0.000000E+00$	$-0.950854E+00$	$0.000000E+00$
4	5	$-0.197725E+01$	$0.220886E+01$	$-0.197861E+01$	$0.220444E+01$
		$-0.197725E+01$	$-0.220886E+01$	$-0.197861E+01$	$-0.220444E+01$
		$-0.219247E+01$	$0.216535E+01$	$-0.282304E+01$	$0.382237E+00$
		$-0.219247E+01$	$-0.216535E+01$	$-0.282304E+01$	$-0.382237E+00$
		$0.464435E+00$	$0.000000E+00$	$-0.201159E+01$	$0.000000E+00$

for example, [28]) by the formula

$$\beta_n^{(j+1)} = \beta_n^{(j)} - \frac{Q(\beta_n^{(j)})}{Q'(\beta_n^{(j)}) - \sum_{k=1}^{n-1} \frac{Q(\beta_n^{(j)})}{\beta_n^{(j)} - \beta_k}}, \tag{5.11}$$

where $\beta_1, \dots, \beta_{n-1}$ are the previously computed zeros of Q. Then $\alpha_1, \dots, \alpha_d$ are computed by the formula $\alpha_n = P(\beta_n)/Q'(\beta_n)$. The derivative $Q'(z)$ is obtained by differentiating the recurrence (5.7).

6. Numerical results. We have implemented the algorithm described in section 5 to compute the representations of σ_n and ω_n through their Laplace transforms. Recall that for the cylinder kernels, σ_n, we have $\nu = n$ while for the sphere kernels, ω_n, we have $\nu = n + 1/2$. Table 1 presents the sizes of the representations for $\varepsilon = 10^{-6}$, 10^{-8}, and 10^{-15} in (4.51). For the cylinder kernels, which are affected by the branch cut, the number of poles for small n is higher than for the sphere kernels. This discrepancy, however, rapidly vanishes as n increases and the asymptotic performance ensues. The $\log(1/\varepsilon)$ dependence of the number of poles for $n \geq 10$ is clear.

For $\varepsilon = 10^{-8}$ we have also computed the maximum norm relative errors which appear in (2.19) by sampling on a fine mesh. For the cylinder kernel with $n = 0$, we expect an $O(1)$ error in a small interval about the origin due to (4.10). However, errors of less than ε are achieved for $|s| > 5 \times 10^{-7}$. This implies a similar accuracy in the approximation of the convolution for times of order 10^6. For all other cases the maximum norm relative errors are of order ε.

Finally, Table 2 presents poles and coefficients for the cylinder kernels for $n = 1, \ldots, 4$ and $\varepsilon = 10^{-6}$ to allow comparison by a reader interested in repeating our calculations. Note that the pole locations are written in terms of $s = z/i$. Extensive tables will be made available on the Web at http://math.nist.gov/mcsd/Staff/BAlpert.

Remark. Our approximate representation of the nonreflecting boundary kernel could be used to reduce the cost of the method introduced by Grote and Keller [8]. The differential operators of degree n obtained in their derivation need only be replaced by the corresponding differential operators of degree $\log n$ for any specified accuracy. It is interesting to note that in the two-dimensional case, where the approach of [8] does not apply, the analysis described above can be used to derive an integrodifferential formulation in the same spirit.

7. Summary. In this paper we have introduced new representations for the logarithmic derivative of a Hankel function of real order, that scale in size as the logarithm of the order. An algorithm to compute the representations was presented and our numerical results demonstrate that the new representations are modest in size for orders and accuracies likely to be of practical interest.

The present motivation for this work is the numerical modeling of nonreflecting boundaries for the wave equation, discussed briefly here and in more detail in [18]. Maxwell's equations are also susceptible to similar treatment as outlined in [29]. The new representations enable the application of the exact nonreflecting boundary conditions, which are global in space and time, to be computationally effective.

8. Appendix: Stability of exact and approximate conditions. In this appendix, we consider the stability of our approach to the design of nonreflecting boundary conditions. Given that we are approximating the exact conditions uniformly, it is natural to expect that our approximations possess similar stability characteristics. This is, indeed, the case. Oddly enough, however, the exact boundary conditions themselves do *not* satisfy the uniform Kreiss–Lopatinski conditions which are necessary and sufficient for strong well-posedness in the usual sense [30]. This may seem paradoxical since the unbounded domain problem itself is strongly well-posed. The difficulty is that the exact reduction of an unbounded domain problem to a bounded domain problem gives rise to forcings (inhomogeneous boundary terms) which live in a restricted subspace. The Kreiss–Lopatinski conditions, on the other hand, require bounds for arbitrary forcings. In that setting, our best estimates result in the loss of 1/3 of a derivative in terms of Sobolev norms. In practice we doubt that this fact is of any significance, and have certainly encountered no stability problems in our long time numerical simulations.

To fill in some of the details, consider a spherical domain Ω of radius one, within which the homogeneous wave equation with homogeneous initial data is satisfied. At the boundary we have

$$\frac{\partial \hat{u}_{nm}}{\partial r} = (1 + \epsilon_n(s)) \frac{s k_n'(s)}{k_n(s)} \hat{u}_{nm} + \hat{g}_{nm}, \tag{8.1}$$

where $\epsilon_n = 0$ for the exact condition and is uniformly small when we use our approximations. Here $\hat{g}_{nm}$ is the spherical harmonic transform of an arbitrary forcing g.

Following Sakamoto, we seek to estimate

$$\mathcal{H}(u) = \int_0^T \left(\|u(\cdot,t)\|_{1,\Omega}^2 + \|u(\cdot,t)\|_{1,\partial\Omega}^2 + \left\| \frac{\partial u}{\partial r}(\cdot,t) \right\|_{0,\partial\Omega}^2 \right) dt, \tag{8.2}$$

where

$$\|f\|_{1,\Omega}^2 = \int_\Omega (f^2 + |\nabla f|^2), \tag{8.3}$$

while $\|\cdot\|_{0,\Omega}$ denotes the usual L_2 norm. On the boundary, $\partial\Omega$, we will make use of fractional Sobolev norms, most easily defined in terms of the spherical harmonic coefficients:

$$\|f\|_{p,\partial\Omega}^2 = \sum_{n,m} (1+n^2)^p |\hat{f}_{nm}|^2. \tag{8.4}$$

Strong well-posedness would follow from showing that

$$\mathcal{H}(u) \le c \int_0^T \|g(\cdot,t)\|_{0,\partial\Omega}^2 dt. \tag{8.5}$$

Instead, we can show that

$$\mathcal{H}(u) \le c \int_0^T \|g(\cdot,t)\|_{1/3,\partial\Omega}^2 dt. \tag{8.6}$$

To prove this, let $s = iz$ and note that

$$k_n(s) \propto h_n^{(1)}(z) \propto z^{-1/2} H_\nu^{(1)}(z), \quad \nu = n + \frac{1}{2}. \tag{8.7}$$

Bounded solutions within the sphere are given by

$$\hat{u}_{nm}(r,s) \propto j_n(rz) \propto (rz)^{-1/2} J_\nu(rz). \tag{8.8}$$

Precisely, setting

$$\hat{u}_{nm}(r,s) = A_{nm}(z)(rz)^{-1/2} J_\nu(rz), \tag{8.9}$$

we find

$$A_{nm}(z) = -\frac{2i}{\pi} z^{1/2} H_\nu^{(1)}(z) \delta_n(z) \hat{g}_{nm}(z), \tag{8.10}$$

where

$$\delta_n = \left(1 - \frac{\pi i}{2}\epsilon_n J_\nu(z)(zH_\nu^{(1)'}(z) - \frac{H_\nu^{(1)}(z)}{2})\right)^{-1}. \tag{8.11}$$

We now estimate norms of the solution. First note that the products in the definition of δ_n, $J_\nu(z)H_\nu^{(1)}(z)$, $zJ_\nu(z)H_\nu^{(1)'}(z)$, are uniformly bounded for $\mathrm{Im}(z) \ge 0$. (See the limits $z \to 0$, $z \to \infty$, and $\nu \to \infty$.) Therefore, as mentioned above, the error term, so long as it's small, has no effect on the estimates we derive, and we simply ignore it. That is, we set $\delta_n = 1$.

We concentrate on the boundary terms in $\mathcal{H}$, as they are both the most straightforward to compute and the most ill behaved. In transform space we have

$$(1+n^2)|\hat{u}_{nm}(1,s)|^2 + |s\hat{u}'_{nm}(1,s)|^2 \le c\gamma_\nu^2(z)|\hat{g}_{nm}(z)|^2, \tag{8.12}$$

$$\gamma_\nu^2(z) = |H_\nu^{(1)}(z)|^2 \left(\nu^2|J_\nu(z)|^2 + |z|^2|J_\nu'(z)|^2\right). \tag{8.13}$$

(Here and throughout, c will denote a positive constant independent of all variables.) We first note that as the only singularities of Bessel functions occur at zero and infinity, we need only consider the limits $z \to 0$, $z \to \infty$, and $\nu \to \infty$. The first two are straightforward:

$$\gamma_\nu^2(z) \approx c\Gamma^2(\nu)(z/2)^{-2\nu}\left(\nu^2(z/2)^{2\nu}/\Gamma^2(\nu+1)\right) = c, \quad z \to 0, \tag{8.14}$$

$$\gamma_\nu^2(z) \approx c|z|^{-1}\left(\nu^2|z|^{-1}\cos^2 z + |z|\sin^2 z\right) \approx c\sin^2 z, \quad z \to \infty. \tag{8.15}$$

For large ν we use the uniform asymptotic expansions of Bessel functions due to Olver [20], which yield

$$\sup_z \gamma_\nu^2(z) = O(\nu^{2/3}). \tag{8.16}$$

From Parseval's relation, we conclude that

$$\int_0^T \left(\|u(\cdot,t)\|^2_{1,\partial\Omega} + \left\|\frac{\partial u}{\partial r}(\cdot,t)\right\|^2_{0,\partial\Omega}\right) dt \le c\int_0^T \|g(\cdot,t)\|^2_{1/3,\partial\Omega}dt. \tag{8.17}$$

The estimation of the spatial integrals is more involved, as for $r < 1$ the solution has two transition zones, $z \approx \nu$ and $rz \approx \nu$, and there are a number of cases to consider. However, the estimates follow along the same lines and lead to the same result.

It is interesting to note that the loss-of-derivative phenomenon is suppressed when one looks at the error due to the approximation of the boundary condition. In that case the transform of the exact solution near the boundary is

$$\frac{h_n^{(1)}(rz)}{h_n^{(1)}(z)}\hat{u}_{nm}(1,s), \tag{8.18}$$

so that the error, e, satisfies the problem above with $\hat{g}_{nm}$ given by

$$\hat{g}_{nm} = \epsilon_n \frac{zh_n^{(1)'}(z)}{h_n^{(1)}(z)}\hat{u}_{nm}(1,s) \equiv \epsilon_n\mu_n(z)\hat{u}_{nm}(1,s). \tag{8.19}$$

Now the best estimate of μ_n takes the form

$$|\mu_n| \le c(|z| + \nu), \tag{8.20}$$

which, in combination with (8.6), would lead to an estimate of the 1-norms of the error in terms of the 4/3-norms of the solution. However, using again the large ν asymptotics, a direct calculation shows

$$|\mu_n\gamma_\nu| \le c(|z| + \nu). \tag{8.21}$$

Thus μ_n is smaller than its maximum by $O(\nu^{-1/3})$ in the transition region where $\gamma_\nu = O(\nu^{1/3})$. Hence we find for the error

$$\mathcal{H}(e) \le c\sup_{n,s}|\epsilon_n|^2 \int_0^T \left(\|u(\cdot,t)\|^2_{1,\partial\Omega} + \left\|\frac{\partial u}{\partial t}(\cdot,t)\right\|^2_{0,\partial\Omega}\right) dt. \tag{8.22}$$

In other words, the 1-norms of the error are controlled by the 1-norms of the solution.

We have, of course, ignored discretization error, which could conceivably cause difficulties in association with the lack of strong well-posedness. To rule them out would require a more detailed analysis. In practice we have encountered no difficulties, even for very long time simulations. We should also note that strong well-posedness could be artificially recovered by perturbing the approximate conditions for large n, allowing high accuracy to be maintained for smooth solutions. Finally, we note that a similar analysis can be carried out in two dimensions.

REFERENCES

[1] D. Givoli, *Non-reflecting boundary conditions*, J. Comput. Phys., 94 (1991), pp. 1–29.

[2] Lu Ting and M. J. Miksis, *Exact boundary conditions for scattering problems*, J. Acoust. Soc. Am., 80 (1986), pp. 1825–1827.

[3] B. Engquist and A. Majda, *Absorbing boundary conditions for the numerical simulation of waves*, Math. Comp., 31 (1977), pp. 629–651.

[4] A. Bayliss and E. Turkel, *Radiation boundary conditions for wave-like equations*, Comm. Pure Appl. Math., 23 (1980), pp. 707–725.

[5] M. Israeli and S. Orszag, *Approximation of radiation boundary conditions*, J. Comput. Phys., 41 (1981), pp. 115–135.

[6] J.-P. Berenger, *A perfectly matched layer for the absorption of electromagnetic waves*, J. Comput. Phys., 114 (1994), pp. 185–200.

[7] I. L. Sofronov, *Conditions for complete transparency on the sphere for the three-dimensional wave equation*, Russian Acad. Sci. Dokl. Math., 46 (1993), pp. 397–401.

[8] M. J. Grote and J. B. Keller, *Nonreflecting boundary conditions for time dependent scattering*, J. Comput. Phys., 52 (1996), p. 127.

[9] T. Hagstrom and S. I. Hariharan, *A formulation of asymptotic and exact boundary conditions using local operators*, Appl. Numer. Math., 27 (1998), pp. 403–416.

[10] T. Hagstrom, S. I. Hariharan, and R. C. MacCamy, *On the accurate long-time solution of the wave equation in exterior domains: Asymptotic expansions and corrected boundary conditions*, Math. Comp., 63 (1994), pp. 507–539.

[11] T. Hagstrom, *On high-order radiation boundary conditions*, in Computational Wave Propagation, B. Engquist and G. Kriegsmann, eds., IMA Vol. Math. Appl. 86, Springer-Verlag, New York, 1996, pp. 23–43.

[12] I. L. Sofronov, *Artificial boundary conditions of absolute transparency for two- and three-dimensional external time-dependent scattering problems*, European J. Appl. Math., 9 (1998), pp. 561–588.

[13] J. R. Driscoll, D. M. Healy, and D. N. Rockmore, *Fast discrete polynomial transforms with applications to data analysis for distance transitive graphs*, SIAM J. Comput., 26 (1997), pp. 1066–1099.

[14] M. J. Mohlenkamp, *A fast transform for spherical harmonics*, J. Fourier Anal. Appl., 5 (1999), pp. 159–184.

[15] J. C. Nédélec, *Quelques proprietes des dérivées logarithmiques des fonctions de hankel*, Technical report 259, Ecole Polytechnique Centre de Mathematiques Appliquees, Palaiseau cedex, France, 1992.

[16] A. Cruz and J. Sesma, *Modulus and phase of the reduced logarithmic derivative of the Hankel function*, Math. Comp., 41 (1983), pp. 597–605.

[17] M. Abramowitz and I. A. Stegun, eds., *Handbook of Mathematical Functions*, 10th printing, National Bureau of Standards, Applied Mathematics Series 55, John Wiley and Sons, New York, 1972.

[18] B. Alpert, L. Greengard, and T. Hagstrom, *Nonreflecting Boundary Conditions for the Time-Dependent Wave Equation*, DOE/CMCL Report 00-001, New York University, 2000.

[19] F. W. J. Olver, *Asymptotics and Special Functions*, Academic Press, New York, 1974.

[20] F. W. J. Olver, *The asymptotic expansion of bessel functions of large order*, Philos. Trans. Roy. Soc. London Ser. A, 247 (1954), pp. 328–368.

[21] F. W. J. Olver, *The asymptotic solution of linear differential equations of the second order for large values of a parameter*, Philos. Trans. Roy. Soc. London Ser. A, 247 (1954), pp. 307–327.

[22] L. GREENGARD AND V. ROKHLIN, *A fast algorithm for particle simulations*, J. Comput. Phys., 73 (1987), pp. 325–348.

[23] C. R. ANDERSON, *An implemenation of the fast multipole method without multipoles*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 923–947.

[24] L. N. TREFETHEN, *Rational Chebyshev approximation on the unit disk*, Numer. Math., 37 (1981), pp. 297–320.

[25] M. H. GUTKNECHT AND L. N. TREFETHEN, *Real and complex Chebyshev approximation on the unit disk and interval*, Bull. Amer. Math. Soc. (New Ser.), 8 (1983), pp. 455–458.

[26] M. H. GUTKNECHT, *Rational Carathéodory-Fejér approximation on a disk, a circle, and an interval*, J. Approx. Theory, 41 (1984), pp. 257–278.

[27] H. P. STARR, *On the Numerical Solution of One-Dimensional Integral and Differential Equations*. Ph.D. thesis, Yale University, New Haven, CT, 1991.

[28] J. STOER AND R. BULIRSCH, *Introduction to Numerical Analysis*, Springer-Verlag, New York, Heidelberg, 1980.

[29] T. HAGSTROM, B. ALPERT, L. GREENGARD, AND S. I. HARIHARAN, *Accurate boundary treatments for Maxwell's equations and their computational complexity*, in the Proceedings of the 14th Annual Review of Progress in Applied Computational Electromagnetics, Monterey, CA, March 16–20, 1998.

[30] R. SAKAMOTO, *Hyperbolic Boundary Value Problems*, Cambridge University Press, Cambridge, UK, 1982.

SIAM J. SCI. COMPUT.
Vol. 20, No. 2, pp. 699–718

© 1998 Society for Industrial and Applied Mathematics

GENERALIZED GAUSSIAN QUADRATURES AND SINGULAR VALUE DECOMPOSITIONS OF INTEGRAL OPERATORS*

N. YARVIN† AND V. ROKHLIN†

Abstract. Generalized Gaussian quadratures appear to have been introduced by Markov late in the last century and have been studied in great detail as a part of modern analysis. They have not been widely used as a computational tool, in part due to an absence of effective numerical schemes for their construction. Recently, a numerical scheme for the design of such quadratures was introduced by Ma et al.; numerical results presented in their paper indicate that such quadratures dramatically reduce the computational cost of the evaluation of integrals under certain conditions. In this paper, we modify their approach, improving the stability of the scheme and extending its range of applicability. The performance of the method is illustrated with several numerical examples.

Key words. quadratures, singular value decompositions, Chebyshev systems, fast algorithms

AMS subject classifications. 65D32, 47G10

PII. S1064827596310779

1. Introduction.

Generalized Gaussian quadratures appear to have been introduced by Markov [11, 12] late in the last century. More recent expositions include those by Krein [9] and Karlin and Studden [8]. Those expositions contain proofs of the existence of such quadratures for wide classes of functions; however, they do not describe a numerical procedure for obtaining the quadrature weights and nodes.

Recently, a paper by Ma, Rokhlin, and Wandzura [10] described a numerical algorithm for obtaining such quadratures. In [10], a version of Newton's method is introduced for the determination of nodes and weights of generalized Gaussian quadratures. The procedure of [10] guarantees the convergence of the Newton algorithm provided it is started sufficiently close to the solution (whose existence is proven in [11, 9, 8]) and utilizes a continuation procedure to provide such starting points. The present paper describes a variation of that algorithm, which consists mainly of two major changes. The first change is that an entirely different continuation scheme is used; with the new continuation scheme, the algorithm is considerably more robust. The second change is the addition of a preprocessing step which, given as input a large class of functions, uses the singular value decomposition (SVD) to produce a set of basis functions suitable for the algorithm.

Since a substantial fraction of the algorithm is changed, this paper is written as a repetition of [10], rather than as a list of changes; however, the portions dealing with quadratures for functions with end-point singularities are omitted.

This paper is organized in the following manner. Section 2 summarizes the necessary material from [9] and [8]. Section 3 briefly describes certain standard numerical tools used by the algorithm. Section 4 contains various analytical results to be used in the construction of the algorithm. Section 5 describes the algorithm in detail. Finally, section 6 contains several numerical examples.

---

[*] Received by the editors October 16, 1996; accepted for publication August 6, 1997; published electronically September 11, 1998. The authors were supported in part by AFOSR grant F49620-93-1-0575, ONR grants N00014-89-J-1527 and N00014-96-1-0188, and a fellowship from the Fannie and John Hertz Foundation.

http://www.siam.org/journals/sisc/x-x/31077.html

[†] Computer Science Department, Yale University, P.O. Box 208285, Yale Station, New Haven, CT 06520-8285 (yarvin@cs.yale.edu, rokhlin@cs.yale.edu).

2. Mathematical preliminaries.

2.1. Chebyshev systems.

DEFINITION 2.1. *A sequence of functions* $\phi_1, \ldots, \phi_n$ *will be referred to as a* Chebyshev system *on the interval* $[a, b]$ *if each of them is continuous and the determinant*

$$\begin{vmatrix} \phi_1(x_1) & \cdots & \phi_1(x_n) \\ \vdots & & \vdots \\ \phi_n(x_1) & \cdots & \phi_n(x_n) \end{vmatrix} \tag{1}$$

is nonzero for any sequence of points $x_1, \ldots, x_n$ *such that* $a \leq x_1 < x_2 \cdots < x_n \leq b$.

An alternate definition of a Chebyshev system is that any linear combination of the functions with nonzero coefficients should have no more than n zeros.

A related definition is that of an extended Chebyshev system.

DEFINITION 2.2. *Given a set of functions* $\phi_1, \ldots, \phi_n$ *which are continuously differentiable on an interval* $[a, b]$, *and given a sequence of points* $x_1, \ldots, x_n$ *such that* $a \leq x_1 \leq x_2 \leq \cdots \leq x_n \leq b$, *let the sequence* $m_1, \ldots, m_n$ *be defined by the formulae*

$$\begin{array}{ll} m_1 = 0, & \\ m_j = 0 & \text{if } j > 1 \text{ and } x_j \neq x_{j-1}, \\ m_j = j - 1 & \text{if } j > 1 \text{ and } x_j = x_{j-1} = \cdots = x_1, \\ m_j = k & \text{if } j > k+1 \text{ and } x_j = x_{j-1} = \cdots = x_{j-k} \neq x_{j-k-1}. \end{array} \tag{2}$$

Let the matrix $C(x_1, \ldots, x_n) = [c_{ij}]$ *be defined by the formula*

$$c_{ij} = \frac{d^{m_j}\phi_i}{dx^{m_j}}(x_j), \tag{3}$$

in which $\frac{d^0\phi_i}{dx^0}(x_j)$ *is taken to be the function value* $\phi_i(x_j)$. *Then* $\phi_1, \ldots, \phi_n$ *will be referred to as an* extended Chebyshev system *on* $[a, b]$ *if the determinant* $|C(x_1, \ldots, x_n)|$ *is nonzero for all such sequences* x_i.

Remark 2.1. It is obvious from Definition 2.2 that an extended Chebyshev system is a special case of the Chebyshev system. The additional constraint is that the successive points x_i at which the function is sampled to form the matrix may be identical; in that case, for each duplicated point, the first corresponding column contains the function values, the second column contains the first derivatives of the functions, the third column contains the second derivatives of the functions, and so forth; this matrix must also be nonsingular.

Examples of Chebyshev and extended Chebyshev systems include the following (additional examples can be found in [8]).

EXAMPLE 2.1. *The powers* $1, x, x^2, \ldots, x^n$ *form an extended Chebyshev system on the interval* $(-\infty, \infty)$.

EXAMPLE 2.2. *The exponentials* $e^{-\lambda_1 x}, e^{-\lambda_2 x}, \ldots, e^{-\lambda_n x}$ *form an extended Chebyshev system for any* $\lambda_1, \ldots, \lambda_n > 0$ *on the interval* $[0, \infty)$.

EXAMPLE 2.3. *The functions* $1, \cos x, \sin x, \cos 2x, \sin 2x, \ldots, \cos nx, \sin nx$ *form a Chebyshev system on the interval* $[0, 2\pi)$.

2.2. Generalized Gaussian quadratures. The quadrature rules considered in this paper are expressions of the form

$$\sum_{j=1}^{n} w_j \phi(x_j), \tag{4}$$

where the points $x_j \in \mathbb{R}$ and coefficients $w_j \in \mathbb{R}$ are referred to as the nodes and weights of the quadrature, respectively. They serve as approximations to integrals of the form

$$\int_a^b \phi(x)\omega(x)dx, \tag{5}$$

where ω has the form

$$\omega(x) = \tilde{\omega}(x) + \sum_{j=1}^{m} \mu_j \cdot \delta(x - \chi_j), \tag{6}$$

with m a nonnegative integer, $\tilde{\omega} : [a,b] \to \mathbb{R}$ an integrable nonnegative function, $\chi_1, \chi_2, \ldots, \chi_m$ points on the interval $[a,b]$, $\mu_1, \mu_2, \ldots, \mu_m$ positive real coefficients, and δ the Dirac δ-function on $\mathbb{R}$.

Remark 2.2. Obviously, (6) defines ω to be a linear combination of a nonnegative function with a finite collection of δ-functions. In a mild abuse of notation, throughout this paper we will be referring to ω as a nonnegative function.

Quadratures are typically chosen so that the quadrature (4) is equal to the desired integral (5) for some set of functions, commonly polynomials of some fixed order. Of these, the classical Gaussian quadrature rules consist of n nodes and integrate polynomials of order $2n - 1$ exactly; these quadratures are used in this paper as a numerical tool (see section 3.2). In [10], the notion of a Gaussian quadrature was generalized as follows.

DEFINITION 2.3. *A quadrature formula will be referred to as* Gaussian *with respect to a set of* $2n$ *functions* $\phi_1, \ldots, \phi_{2n} : [a,b] \to \mathbb{R}$ *and a weight function* $\omega : [a,b] \to \mathbb{R}^+$, *if it consists of* n *weights and nodes, and integrates the functions* ϕ_i *exactly with the weight function* ω *for all* $i = 1, \ldots, 2n$. *The weights and nodes of a Gaussian quadrature will be referred to as Gaussian weights and nodes, respectively.*

The following theorem appears to be due to Markov [11, 12]; proofs of it can also be found in [9] and [8] (in a somewhat different form).

THEOREM 2.1. *Suppose that the functions* $\phi_1, \ldots, \phi_{2n} : [a,b] \to \mathbb{R}$ *form a Chebyshev system on* $[a,b]$. *Suppose in addition that* $\omega : [a,b] \to \mathbb{R}$ *is defined by* (6), *and that either*

$$\int_a^b \tilde{\omega}(x)dx > 0 \tag{7}$$

or $m \geq n$ *(or both). Then there exists a unique Gaussian quadrature for* $\phi_1, \ldots, \phi_{2n}$ *on* $[a,b]$ *with respect to the weight function* ω. *The weights of this quadrature are positive.*

2.3. Total positivity.

A concept closely related to that of an extended Chebyshev system is that of a extended totally positive (ETP) kernel.

DEFINITION 2.4. *Given a function* $K : [a,b] \times [c,d] \to \mathbb{R}$ *which is* n *times continuously differentiable, and given a sequence of points* $x_1, \ldots, x_n$ *such that* $c \leq x_1 \leq x_2 \leq \cdots \leq x_n \leq d$, *let the sequence* $m_1, \ldots, m_n$ *be defined by* (2). *Let the functions* $\phi_1, \ldots, \phi_n$ *be defined by the formula*

$$\phi_j(t) = \frac{\partial^{m_j} K}{\partial x^{m_j}}(x_j, t), \tag{8}$$

in which $\frac{\partial^0 K}{\partial x^0}(x_j, t)$ *is taken to be the function value* $K(x_j, t)$. *Then* K *will be referred to as* ETP *if the functions* $\phi_1, \dots, \phi_n$ *form an extended Chebyshev system on* $[c, d]$ *for all such sequences of* x_i.

Examples of ETP kernels include the following (additional examples can be found in [8]).

EXAMPLE 2.4. *The function* e^{-xt} *is ETP for* $x, t \in [0, \infty)$.

EXAMPLE 2.5. *The function* $e^{-(x-t)^2}$ *is ETP for* $x, t \in (-\infty, \infty)$.

EXAMPLE 2.6. *The function* $1/(x + t)$ *is ETP for* $x, t \in (0, \infty)$.

A proof of the following lemma can be found in [8], for example.

LEMMA 2.2. *Suppose that* K *and* L *are ETP functions of two variables. Then the function* M *defined by the formula*

$$M(x,t) = \int_c^d K(x,s)L(s,t)ds \tag{9}$$

is ETP. In other words, if the kernels of two integral operators are ETP, the kernel of the product of the two operators is ETP.

The following theorem can be found in [7, 8].

THEOREM 2.3. *Suppose that* $K : [a, b] \times [a, b] \to \mathbb{R}$ *is an ETP kernel. Then the first* p *eigenfunctions of the integral operator* $T : L^2[a, b] \to L^2[a, b]$ *defined by the formula*

$$(T\phi)(x) = \int_a^b K(x,s)\phi(s)ds \tag{10}$$

constitute an extended Chebyshev system for any $p \geq 1$.

3. Numerical preliminaries.

3.1. Newton's method.

In this section we discuss two well-known numerical techniques: Newton's method and the continuation method. A more detailed discussion of these techniques can be found, for example, in [14].

Newton's method is an iterative method for the solution of nonlinear systems of equations of the form $F(x) = 0$, where $F : \mathbb{R}^n \to \mathbb{R}^n$ is a continuously differentiable function of the form

$$F(x) = \begin{pmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{pmatrix}, \tag{11}$$

and $x = (x_1, \dots, x_n)^T$. The method uses the Jacobian matrix J of F, which is defined by the formula

$$J(x) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x) & \cdots & \frac{\partial f_1}{\partial x_n}(x) \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1}(x) & \cdots & \frac{\partial f_n}{\partial x_n}(x) \end{pmatrix}. \tag{12}$$

LEMMA 3.1 (Newton's method). *Suppose that for some* $y \in \mathbb{R}^n$,

$$F(y) = 0, \tag{13}$$

with $F : \mathbb{R}^n \to \mathbb{R}^n$ *defined by* (11), *and that* $|J(y)| \neq 0$, *with* $|J(y)|$ *denoting the determinant of the matrix* $J(x)$ *defined in* (12), *evaluated at the point* y. *Given a starting point* $y_0 \in \mathbb{R}^n$, *let the sequence* $y_1, y_2, \dots$ *be defined by the formula*

$$y_{k+1} = y_k - J^{-1}(y_k)F(y_k). \tag{14}$$

Then there exists a positive real number ε *such that for any* y_0 *satisfying the inequality* $\|y_0 - y\| < \varepsilon$, *the sequence* (14) *converges to* y *quadratically; that is, there exists a positive real number* α *such that*

$$\|y_{k+1} - y\| < \alpha \|y_k - y\|^2. \tag{15}$$

3.1.1. Continuation method. In order for Newton's method to converge, the starting point which is provided to it must be close to the desired solution. One scheme for generating such starting points is the continuation method, which is as follows.

Suppose that in addition to the function $F : \mathbb{R}^n \to \mathbb{R}^n$ whose zero is to be found, another function $G : [0,1] \times \mathbb{R}^n \to \mathbb{R}^n$ is available which possesses the following properties.

(i) For any $x \in \mathbb{R}^n$,

$$G(1, x) = F(x). \tag{16}$$

(ii) The solution of the equation $G(0, x) = 0$ is known.

(iii) For all $t \in [0,1]$, the equation $G(t, x) = 0$ has a unique solution x such that the conditions of Lemma 3.1 are satisfied.

(iv) The solution x is a continuous function of t.

If these conditions are met, an algorithm for the solution of $F(x) = 0$ is as follows. Let the points t_i, for $i = 1, \dots, m$, be defined by the formula $t_i = i/m$. Solve in succession the equations

$$\begin{gathered} G(t_1, x) = 0, \\ G(t_2, x) = 0, \\ \vdots \\ G(t_m, x) = 0, \end{gathered}$$

using Newton's method, with the starting point for Newton's method for each equation taken to be the solution of the preceding equation. The solution x of the final equation $G(t_m, x) = 0$ is, by (16), identical to the solution of the desired equation $F(x) = 0$. Obviously, for sufficiently large m, Newton's method is guaranteed by Lemma 3.1 to converge at each step.

Remark 3.1. In practice, it is desirable to choose the smallest m for which the above algorithm will work, in order to reduce the computational cost of the scheme. On the other hand, the largest step $(t_i - t_{i-1})$ for which the Newton method will converge commonly varies as a function of t. Thus the algorithm described in this paper uses an adaptive version of the scheme.

3.2. Gaussian integration and interpolation. Classical Gaussian quadrature rules are a well-known numerical tool (see, for instance, [14]); they integrate polynomials of order $2n - 1$ exactly with respect to some weight function and consist of n weights and nodes. A variety of Gaussian quadratures were analyzed in the last century, each being defined by a distinct weight function. Of these, the algorithm presented in this paper uses only the Gaussian quadratures for the weight function $\omega(x) = 1$ on the region of integration $[-1, 1]$. These quadratures are closely associated with the Legendre polynomials; we will refer to their nodes as Legendre nodes.

Another numerical tool used in this paper is polynomial interpolation on Legendre nodes. Interpolation refers to the following problem: given two finite real sequences $f_1, \ldots, f_n \in \mathbb{R}$ and $x_1, \ldots, x_n \in [a, b]$, construct a function $f : [a, b] \to \mathbb{R}$ such that $f(x_i) = f_i$ for all $i = 1, \ldots, n$. One interpolation scheme is polynomial interpolation, in which the interpolating function f is a polynomial of degree $n-1$. As is well known, such a polynomial always exists and is unique. However, in general two numerical difficulties arise with polynomial interpolation using polynomials of high order. The first is that for many sequences of points $\{x_i\}$, the values of the interpolating polynomial between the points $\{x_i\}$ are not well conditioned as a function of the values $\{f_i\}$ to be interpolated. The second is that even for those sequences of points where the computation of the values of the interpolating polynomial is well conditioned, the computation of the coefficients of the power series of the interpolating polynomial is extremely ill conditioned.

As is well known, these difficulties do not arise if the points $\{x_i\}$ are taken to be Chebyshev nodes and the interpolating polynomial is computed as a series of Chebyshev polynomials rather than as a power series. As the following lemma shows, the difficulties also do not arise if the points $\{x_i\}$ are taken to be Legendre nodes and the interpolating polynomial is computed as a series of Legendre polynomials. The lemma makes use of the following properties of the Legendre polynomials: first, that the ith Legendre polynomial P_i has degree i; second, that the polynomials P_i form an orthonormal system of functions on $[-1, 1]$.

LEMMA 3.2. *Suppose that* $x_1, \ldots, x_n \in [-1, 1]$ *are the Legendre nodes of order* n, *and that* $w_1, \ldots, w_n \in \mathbb{R}$ *are the associated Gaussian weights. Given a sequence* $f_1, \ldots, f_n \in \mathbb{R}$, *let* $p : [-1, 1] \to \mathbb{R}$ *be the interpolating polynomial of degree* $n-1$ *such that* $p(x_i) = f_i$ *for all* $i = 1, \ldots, n$, *and let* $c_0, \ldots, c_{n-1}$ *be the coefficients of the Legendre series of* p; *that is,*

$$p(x) = \sum_{i=0}^{n-1} c_i P_i(x), \tag{17}$$

where $P_i(x)$ *is the* i*th Legendre polynomial. Then the following relation holds:*

$$\sum_{i=1}^{n} w_i f_i^2 = \int_{-1}^{1} p(x)^2 dx = \sum_{i=0}^{n-1} c_i^2. \tag{18}$$

Proof. The second equality of (18) follows from (17) and the orthonormality of the Legendre polynomials. The first equality may be proven as follows: the polynomial p has degree $n-1$; thus its square has degree $2n-2$. Since the Gaussian quadrature integrates exactly all polynomials up to order $2n-1$, it integrates p^2 exactly; thus the first equality of (18) holds. □

3.3. Singular value decomposition. The singular value decomposition (SVD) is a ubiquitous tool in numerical analysis, which is given for the case of real matrices by the following lemma (see, for instance, [3] for more details).

LEMMA 3.3. *For any* $n \times m$ *real matrix* A, *there exists an* $n \times p$ *real matrix* U *with orthonormal columns, an* $m \times p$ *real matrix* V *with orthonormal columns, and a*

$p \times p$ real diagonal matrix $S = [s_{ij}]$ whose diagonal entries are nonnegative, such that $A = USV^*$ and that $s_{ii} \geq s_{i+1,i+1}$ for all $i = 1, \ldots, p-1$.

The diagonal entries s_{ii} of S are called singular values; the columns of the matrix V are called right singular vectors; the columns of the matrix U are called left singular vectors.

3.4. Singular value decompositions of integral operators. This section, which follows [5], contains an existence theorem for a factorization of integral operators. The operators $T : L^2[c,d] \to L^2[a,b]$ to which it applies are of the form

$$(Tf)(x) = \int_c^d K(x,t)f(t)dt, \tag{19}$$

in which the function $K : [a,b] \times [c,d] \to \mathbb{R}$ is referred to as the kernel of the operator T. Throughout this section, it will be assumed that all functions are square integrable; the term "norm" will mean the L^2 norm.

The following theorem, which defines the factorization, is proven in a more general form as Theorem VI.17 in [13].

THEOREM 3.4. *Suppose that the function $K : [a,b] \times [c,d] \to \mathbb{R}$ is square integrable. Then there exist two orthonormal sequences of functions $u_i : [a,b] \to \mathbb{R}$ and $v_i : [c,d] \to \mathbb{R}$ and a sequence $s_i \in \mathbb{R}$, for $i = 1, \ldots, \infty$, such that*

$$K(x,t) = \sum_{i=1}^{\infty} u_i(x)s_iv_i(t) \tag{20}$$

and that $s_1 \geq s_2 \geq \cdots \geq 0$. The sequence s_i is uniquely determined by K. Furthermore, the functions v_i are eigenfunctions of the operator T^*T, where T is defined by (19)*, and the values s_i are the square roots of the eigenvalues of T^*T.*

By analogy to the finite-dimensional case, we will refer to this factorization as the singular value decomposition. We will refer to the functions u_i as left singular functions of K (or of T), to v_i as right singular functions, and to s_i as singular values.

As is the case for the discrete SVD, this decomposition can be used to construct an approximation to the function K by discarding small singular values and the associated singular functions:

$$K(x,t) \simeq \sum_{i=1}^{p} u_i(x)s_iv_i(t). \tag{21}$$

The error of this approximation can then be computed from (20):

$$K(x,t) - \sum_{i=1}^{p} u_i(x)s_iv_i(t) = \sum_{i=p+1}^{\infty} u_i(x)s_iv_i(t), \tag{22}$$

and, therefore,

$$\left\| K(x,t) - \sum_{i=1}^{p} u_i(x)s_iv_i(t) \right\| = \sqrt{\sum_{i=p+1}^{\infty} s_i^2}. \tag{23}$$

Using (21), integrals of the form

$$\int_a^b K(x,t)\omega(x)dx \tag{24}$$

can be approximated by the formula

$$\int_a^b K(x,t)\omega(x)dx \simeq \int_a^b \sum_{i=1}^{p} u_i(x)s_i v_i(t)\omega(x)dx$$

$$\simeq \sum_{i=1}^{p} s_i v_i(t) \int_a^b u_i(x)\omega(x)dx. \tag{25}$$

Thus a quadrature which is exact for each of the integrals

$$\int_a^b u_i(x)\omega(x)dx, \tag{26}$$

for $i = 1, \ldots, p$, is an approximate quadrature for integrals of the form (24).

Many properties of the singular functions of an integral operator can be deduced from the corresponding properties of eigenfunctions of integral operators; a property of concern in this paper is that of forming an extended Chebyshev system and is addressed by the following theorem.

THEOREM 3.5. *Suppose that* $K : [a,b] \times [c,d] \to \mathbb{R}$ *is ETP. Then the first p left singular functions of K form an extended Chebyshev system for any p; likewise the first p right singular functions of K form an extended Chebyshev system for any p.*

Proof. Let the integral operator $T : L^2[c,d] \to L^2[a,b]$ be defined by the formula

$$(Tf)(x) = \int_c^d K(x,t)f(t)dt, \tag{27}$$

and let the function $L : [a,b] \to [a,b]$ be defined by the formula

$$L(x,t) = \int_c^d K(x,s)K(t,s)ds. \tag{28}$$

Clearly, the integral operator $S : L^2[a,b] \to L^2[a,b]$ defined by the formula $S = T^*T$ has the kernel L:

$$(S\phi)(x) = \int_a^b \int_c^d K(x,s)K(t,s)ds\phi(t)dt$$

$$= \int_a^b L(x,t)\phi(t)dt. \tag{29}$$

Since K is ETP, L is also ETP, due to Lemma 2.2. Thus by Theorem 2.3, the eigenfunctions of S constitute an extended Chebyshev system. By Theorem 3.4, these eigenfunctions are identical to the left singular functions of T, which proves that the first p left singular functions of T constitute an extended Chebyshev system for any p. The proof for the right singular functions is identical. $\square$

4. Analytical apparatus.

4.1. Convergence of Newton's method. In this section, we observe that the nodes and the weights of a Gaussian quadrature satisfy a certain system of nonlinear equations. We then prove that the Newton method for this system of equations is always quadratically convergent, provided the functions to be integrated constitute an extended Chebyshev system.

Given a set of functions $\phi_1, \ldots, \phi_{2n}$ and a weight function ω, the Gaussian quadrature is defined by the system of equations

$$\sum_{j=1}^{n} w_j \phi_1(x_j) = \int_a^b \phi_1(x)\omega(x)dx,$$

$$\sum_{j=1}^{n} w_j \phi_2(x_j) = \int_a^b \phi_2(x)\omega(x)dx,$$

$$\vdots$$

$$\sum_{j=1}^{n} w_j \phi_{2n}(x_j) = \int_a^b \phi_{2n}(x)\omega(x)dx \tag{30}$$

(see Definition 2.3). Let the left-hand sides of these equations be denoted by f_1 through f_{2n}. Then each f_i is a function of the weights $w_1, \ldots, w_n$ and nodes $x_1, \ldots, x_n$ of the quadrature. Its partial derivatives are given by the obvious formulae

$$\frac{\partial f_k}{\partial w_i} = \phi_k(x_i), \tag{31}$$

$$\frac{\partial f_k}{\partial x_i} = w_i \phi_k'(x_i). \tag{32}$$

Thus the Jacobian matrix of the system (30) is

$$J(x_1, \ldots, x_n, w_1, \ldots, w_n) = \begin{pmatrix} \phi_1(x_1) & \cdots & \phi_1(x_n) & w_1\phi_1'(x_1) & \cdots & w_n\phi_1'(x_n) \\ \vdots & & \vdots & \vdots & & \vdots \\ \phi_{2n}(x_1) & \cdots & \phi_{2n}(x_n) & w_1\phi_{2n}'(x_1) & \cdots & w_n\phi_{2n}'(x_n) \end{pmatrix}. \tag{33}$$

LEMMA 4.1. *Suppose that the functions* $\phi_1, \ldots, \phi_{2n}$ *form an extended Chebyshev system. Let the Gaussian quadrature for these functions be denoted by* $\hat{w}_i$ *and* $\hat{x}_i$. *Then the determinant of* J *is nonzero at the point which constitutes the Gaussian quadrature; in other words,* $|J(\hat{x}_1, \ldots, \hat{x}_n, \hat{w}_1, \ldots, \hat{w}_n)| \neq 0$.

Proof. It is immediately obvious from (33) that

$$|J(\hat{x}_1, \ldots, \hat{x}_n, \hat{w}_1, \ldots, \hat{w}_n)| = \hat{w}_1 \cdot \hat{w}_2 \cdot \cdots \cdot \hat{w}_{n-1} \cdot \hat{w}_n \cdot \begin{vmatrix} \phi_1(\hat{x}_1) & \cdots & \phi_1(\hat{x}_n) & \phi_1'(\hat{x}_1) & \cdots & \phi_1'(\hat{x}_n) \\ \vdots & & \vdots & \vdots & & \vdots \\ \phi_{2n}(\hat{x}_1) & \cdots & \phi_{2n}(\hat{x}_n) & \phi_{2n}'(\hat{x}_1) & \cdots & \phi_{2n}'(\hat{x}_n) \end{vmatrix}. \tag{34}$$

If $\phi_1, \ldots, \phi_{2n}$ form an extended Chebyshev system, then by Theorem 2.1 the weights $\hat{w}_1, \ldots, \hat{w}_n$ of the Gaussian quadrature are positive. In addition, by the definition of an extended Chebyshev system, the determinant in the right-hand side of (34) is nonzero. Thus

$$|J(\hat{x}_1, \ldots, \hat{x}_n, \hat{w}_1, \ldots, \hat{w}_n)| \neq 0. \quad \square \tag{35}$$

Using the inverse function theorem, we immediately obtain the following corollary.

COROLLARY 4.2. *Under the conditions of Lemma* 4.1, *the Gaussian weights and nodes depend continuously on the weight function.*

4.2. Interpolation. This section contains two basic lemmas about interpolation. The following lemma shows that any interpolation scheme on an interval $[a,b]$ whose output depends linearly on its input is characterized by a finite sequence of functions $[a,b] \to \mathbb{R}$.

LEMMA 4.3. *Suppose* $L : \mathbb{R}^n \to L^2[a,b]$ *is an interpolation scheme with* n *nodes* $x_1, \ldots, x_n \in [a,b]$, *and that* L *is a linear mapping. Then there exists a sequence of functions* $\alpha_1, \ldots, \alpha_n : [a,b] \to \mathbb{R}$ *such that for any vector* $f \in \mathbb{R}^n$, *with elements* $f = (f_1, \ldots, f_n)^T$,

$$(Lf)(x) = \sum_{i=1}^{n} f_i \alpha_i(x) \tag{36}$$

for all $x \in [a,b]$.

Proof. Let the vectors $e_1, \ldots, e_n \in \mathbb{R}^n$ with elements $e_i = (e_{i1}, \ldots, e_{in})^T$ be the standard basis in $\mathbb{R}^n$; that is, $e_{ii} = 1$ for all $i = 1, \ldots, n$, and $e_{ij} = 0$ for all $i, j = 1, \ldots, n$ such that $i \neq j$. Let the functions $\alpha_1, \ldots, \alpha_n : [a,b] \to \mathbb{R}$ be defined by the formula $\alpha_i = Le_i$. Since the interpolation scheme L is linear, for any vector $f \in \mathbb{R}^n$ with elements $f = (f_1, \ldots, f_n)^T$, and for any point $x \in [a,b]$,

$$\begin{aligned}(Lf)(x) &= \left(L\left(\sum_{i=1}^{n} f_i e_i\right)\right)(x)\\ &= \sum_{i=1}^{n} f_i (Le_i)(x)\\ &= \sum_{i=1}^{n} f_i \alpha_i(x). \qquad \square \end{aligned} \tag{37}$$

In the case of polynomial interpolation, the functions α_i are referred to as Lagrange polynomials; by analogy to that case, we will in general refer to the functions α_i as the Lagrange functions of the interpolation scheme.

The following lemma provides an error bound for approximation of a function of two variables using two one-dimensional interpolation formulae, expressed in terms of error bounds for each one-dimensional interpolation scheme applied separately. Its proof is an exercise in elementary analysis and is omitted.

LEMMA 4.4. *Suppose that* $x_1, x_2, \ldots, x_n \in [a,b]$ *and* $t_1, t_2, \ldots, t_m \in [c,d]$ *are two finite real sequences, and that* $\alpha_1, \alpha_2, \ldots, \alpha_n : [a,b] \to \mathbb{R}$ *and* $\beta_1, \beta_2, \ldots, \beta_m : [c,d] \to \mathbb{R}$ *are two sequences of bounded functions. Suppose further that* $L_1 : \mathbb{R}^n \to L^\infty[a,b]$ *is an interpolation formula with the nodes* $x_1, \ldots, x_n$ *and Lagrange functions* $\alpha_1, \ldots, \alpha_n$, *and* $L_2 : \mathbb{R}^m \to L^\infty[c,d]$ *is an interpolation formula with the nodes* $t_1, \ldots, t_m$ *and Lagrange functions* $\beta_1, \ldots, \beta_m$. *Suppose that* $\eta \in \mathbb{R}$ *is such that*

$$\sum_{i=1}^{n} |\alpha_i(x)| < \eta \tag{38}$$

for all $x \in [a,b]$, *or*

$$\sum_{j=1}^{m} |\beta_j(t)| < \eta \tag{39}$$

for all $t \in [c,d]$. *Finally, suppose that* K *is a function* $[a,b] \times [c,d] \to \mathbb{R}$, *and that for all* $x \in [a,b]$ *and* $t \in [c,d]$,

$$\left| K(x,t) - \sum_{i=1}^{n} K(x_i,t)\alpha_i(x) \right| < \varepsilon \tag{40}$$

and

$$\left| K(x,t) - \sum_{j=1}^{m} K(x,t_j)\beta_j(t) \right| < \varepsilon. \tag{41}$$

Then

$$\left| K(x,t) - \sum_{i=1}^{n}\sum_{j=1}^{m} K(x_i,t_j)\alpha_i(x)\beta_j(t) \right| < \varepsilon(1+\eta) \tag{42}$$

for all $x \in [a,b]$ *and* $t \in [c,d]$.

4.3. Approximation of SVD of an integral operator. This section describes a numerical procedure for computing an approximation to the SVD of an integral operator.

The algorithm uses quadratures which possess the following property.

DEFINITION 4.1. *We will say that the combination of a quadrature and an interpolation scheme* preserves inner products *on an interval* $[a,b]$ *if it possesses the following properties.*

(i) *The nodes of the quadrature are identical to the nodes of the interpolation scheme.*

(ii) *The function which is output by the interpolation scheme depends in a linear fashion on the values input to the interpolation scheme.*

(iii) *The quadrature integrates exactly any product of two interpolated functions; that is, for any two functions* $f,g : [a,b] \to \mathbb{R}$ *produced by the interpolation scheme, the integral*

$$\int_a^b f(x)g(x)dx \tag{43}$$

is computed exactly by the quadrature.

Quadratures and interpolation schemes which possess this property include the following.

EXAMPLE 4.1. *The combination of a (classical) Gaussian quadrature at Legendre nodes and polynomial interpolation at the same nodes preserves inner products, since polynomial interpolation on* n *nodes produces an interpolating polynomial of order* $n-1$, *the product of two such polynomials is a polynomial of order* $2n-2$, *and a Gaussian quadrature integrates exactly all polynomials up to order* $2n-1$.

EXAMPLE 4.2. *If an interval is broken into several subintervals, and a quadrature and interpolation scheme which preserves inner products is used on each subinterval, then the arrangement as a whole preserves inner products on the original interval. (This follows directly from the definition.)*

EXAMPLE 4.3. *The combination of the trapezoidal rule on the interval* $[0,2\pi]$ *and Fourier interpolation (using the interpolation functions* 1, $\cos x$, $\sin x$, $\cos 2x$, $\sin 2x$, $\ldots$, $\cos nx$, $\sin nx$*) preserves inner products.*

The algorithm takes as input a function $K : [a, b] \times [c, d] \to \mathbb{R}$. It uses the following numerical tools:

(i) A quadrature and an interpolation scheme on the interval $[a, b]$ which preserve inner products. Let the weights and nodes of this quadrature be denoted by $w_1^x, \ldots, w_n^x \in \mathbb{R}$ and $x_1, \ldots, x_n \in [a, b]$, respectively. Let the Lagrange functions (see section 4.2) of the interpolation scheme be denoted by $\alpha_1, \ldots, \alpha_n : [a, b] \to \mathbb{R}$.

(ii) A quadrature and an interpolation scheme on the interval $[c, d]$ which preserve inner products. Let the weights and nodes of this quadrature be denoted by $w_1^t, \ldots, w_m^t \in \mathbb{R}$ and $t_1, \ldots, t_m \in [c, d]$, respectively. Let the Lagrange functions of the interpolation scheme be denoted by $\beta_1, \ldots, \beta_m : [c, d] \to \mathbb{R}$.

As will be shown below, the accuracy of the algorithm is then determined by the accuracy to which the above two interpolation schemes approximate K.

The output of the algorithm is a sequence of functions $u_1, \ldots, u_p : [a, b] \to \mathbb{R}$, a sequence of functions $v_1, \ldots, v_p : [c, d] \to \mathbb{R}$, and a sequence of singular values $s_1, \ldots, s_p \in \mathbb{R}$, which form an approximation to the SVD of K.

Description of the algorithm.

(i) Construct the $n \times m$ matrix $A = [a_{ij}]$ defined by the formula

$$a_{ij} = K(x_i, t_j)\sqrt{w_i^x \cdot w_j^t}. \tag{44}$$

(ii) Compute the SVD of A to produce the factorization

$$A = USV^*, \tag{45}$$

where $U = [u_{ij}]$ is an $n \times p$ matrix with orthonormal columns, $V = [v_{ij}]$ is an $m \times p$ matrix with orthonormal columns, and S is a $p \times p$ diagonal matrix whose jth diagonal entry is s_j.

(iii) Construct the $n \times p$ matrix $\hat{U} = [\hat{u}_{ij}]$ and the $m \times p$ matrix $\hat{V} = [\hat{v}_{ij}]$ defined by the formulae

$$\hat{u}_{ik} = u_{ik}/\sqrt{w_i^x}, \tag{46}$$

$$\hat{v}_{jk} = v_{jk}/\sqrt{w_j^t}. \tag{47}$$

(iv) For any points $x \in [a, b]$ and $t \in [c, d]$, evaluate the functions $u_k : [a, b] \to \mathbb{R}$ and $v_k : [c, d] \to \mathbb{R}$ via the formulae

$$u_k(x) = \sum_{i=1}^{n} \hat{u}_{ik} \cdot \alpha_i(x), \tag{48}$$

$$v_k(t) = \sum_{j=1}^{m} \hat{v}_{jk} \cdot \beta_j(t), \tag{49}$$

for all $k = 1, \ldots, p$.

THEOREM 4.5. *Suppose that the combination of the quadrature with weights and nodes* $w_1^x, \ldots, w_n^x \in \mathbb{R}$ *and* $x_1, \ldots, x_n \in [a, b]$, *respectively, and the interpolation scheme with Lagrange functions* $\alpha_1, \ldots, \alpha_n : [a, b] \to \mathbb{R}$, *preserves inner products on* $[a, b]$.

Suppose in addition that the combination of the quadrature with weights and nodes $w_1^t, \ldots, w_m^t \in \mathbb{R}$ *and* $t_1, \ldots, t_m \in [c, d]$, *respectively, and the interpolation scheme with Lagrange functions* $\beta_1, \ldots, \beta_m : [c, d] \to \mathbb{R}$, *preserves inner products on* $[c, d]$.

For any function $K : [a, b] \times [c, d] \to \mathbb{R}$, *let* $u_i : [a, b] \to \mathbb{R}$, $v_i : [c, d] \to \mathbb{R}$, *and* $s_i \in \mathbb{R}$ *be defined in* (44)–(49), *for all* $i = 1, \ldots, p$. *Then*

(i) *The functions* u_i *are orthonormal, i.e.,*

$$\int_a^b u_i(x)u_k(x)dx = \delta_{ik} \tag{50}$$

for all $i, k = 1, \ldots, p$, *with* δ_{ik} *the Kronecker symbol* ($\delta_{ij} = 1$ *if* $i = j$, 0 *otherwise).*

(ii) *The functions* v_i *are orthonormal, i.e.,*

$$\int_c^d v_i(t)v_k(t)dx = \delta_{ik} \tag{51}$$

for all $i, k = 1, \ldots, p$.

(iii) *The function* $\tilde{K} : [a, b] \times [c, d] \to \mathbb{R}$ *defined by the formula*

$$\tilde{K}(x,t) = \sum_{j=1}^{p} s_j u_j(x) v_j(t) \tag{52}$$

is identical to the function produced by sampling K *on the grid of points* (x_i, t_j), *then interpolating with the two interpolation schemes. That is,*

$$\tilde{K}(x,t) = \sum_{i=1}^{n}\sum_{j=1}^{m} K(x_i, t_j)\alpha_i(x)\beta_j(t). \tag{53}$$

Proof. We first prove (53). Combining (48), (49), and (52), we have

$$\begin{aligned}
\tilde{K}(x,t) &= \sum_{k=1}^{p} s_k \left(\sum_{i=1}^{n} u_k(x_i)\alpha_i(x)\right)\left(\sum_{j=1}^{m} v_k(w_j^x)\beta_j(t)\right) \\
&= \sum_{i=1}^{n}\sum_{j=1}^{m}\left(\sum_{k=1}^{p} u_k(x_i) s_k v_k(w_j^x)\right)\alpha_i(x)\beta_j(t) \\
&= \sum_{i=1}^{n}\sum_{j=1}^{m}\left(\sum_{k=1}^{p} (u_{ik}/\sqrt{w_i^x}) s_k (v_{jk}/\sqrt{w_j^t})\right)\alpha_i(x)\beta_j(t) \\
&= \sum_{i=1}^{n}\sum_{j=1}^{m}\left(\sum_{k=1}^{p} u_{ik} s_k v_{jk}/\sqrt{w_i^x w_j^t}\right)\alpha_i(x)\beta_j(t) \\
&= \sum_{i=1}^{n}\sum_{j=1}^{m}\left(a_{ij}/\sqrt{w_i^x w_j^t}\right)\alpha_i(x)\beta_j(t). \qquad (54)
\end{aligned}$$

Now (53) follows from the combination of (54) and (44).

We now demonstrate the orthonormality of the functions u_i. Since these are functions produced by interpolation, and since the quadrature on $[a, b]$ is assumed to integrate exactly all products of pairs of interpolated functions,

$$\begin{aligned}
\int_a^b u_i(x)u_k(x)dx &= \sum_{j=1}^{n} w_j^x u_i(x_j) u_k(x_j) \\
&= \sum_{j=1}^{n} w_j^x (u_{ji}/\sqrt{w_j^x})(u_{jk}/\sqrt{w_j^x}) \\
&= \sum_{j=1}^{n} u_{ji} u_{jk}. \qquad (55)
\end{aligned}$$

Since the last sum in (55) is the inner product of two columns of the orthonormal matrix U (see (45)),

$$\int_a^b u_i(x)u_k(x)dx = \delta_{ik}. \tag{56}$$

The orthonormality of the functions v_i is proven in the same manner. □

Remark 4.1. Obviously, the above proof approximates the SVD of the operator $T : L^2[c,d] \to L^2[a,b]$ with the kernel K by constructing an approximation $\tilde{T}$ with kernel $\tilde{K}$ to the operator T that is of finite rank, and constructing the exact SVD of the latter.

OBSERVATION 4.2. *In the preceding proof, the assumption that each combination of quadrature and interpolation scheme preserves inner products was used only to demonstrate the orthonormality of the corresponding singular functions. Thus if the conditions of Theorem* 4.5 *hold, with the exception that the quadrature on* $[a,b]$ *does not preserve inner products, then* (51) *and* (53) *hold (but, in general,* (50) *does not).*

Remark 4.3. Theorem 4.5 and Lemma 4.4 generalize trivially to higher dimensions. One-dimensional quadratures and interpolation formulae have to be replaced with their multidimensional counterparts; otherwise, the proofs are unchanged.

5. Numerical algorithm.

This section describes a numerical algorithm for the evaluation of nodes and weights of generalized Gaussian quadratures. The algorithm's input is a sequence of functions $\phi_1, \ldots, \phi_{2n} : [a,b] \to \mathbb{R}$ which form an extended Chebyshev system on $[a,b]$, and a weight function $\omega_1 : [a,b] \to \mathbb{R}^+$. Its output is the weights and nodes of the quadrature. The main components of the algorithm are as follows (not listed in order of execution).

(i) Newton's method is used to solve (30) which defines the Gaussian quadrature.

(ii) An adaptive version of the continuation method (section 3.1.1) is used to provide starting points for Newton's method. The continuation scheme used here is different from that used in [10]; the details of the continuation scheme and of the method of adaption are described below.

(iii) The algorithm of section 4.3 can be used as an optional preprocessing step, which takes as input a kernel of an integral operator and produces its singular functions. The first $2n$ of the left singular functions are then used as input to the main algorithm.

5.1. Continuation scheme.

The continuation scheme used is as follows. Let the weight functions $\omega : [0,1] \times [a,b] \to \mathbb{R}^+$ be defined by the formula

$$\omega(\alpha, x) = \alpha\omega_1(x) + (1-\alpha)\sum_{j=1}^{n} \delta(x - c_j), \tag{57}$$

where ω_1 is the weight function for which a Gaussian quadrature is desired, δ denotes the Dirac delta function, and the points $c_j \in [a,b]$ are arbitrary distinct points. These weight functions have the following properties.

(i) With $\alpha = 1$, the weight function is equal to the desired weight function ω_1, due to (57).

(ii) With $\alpha = 0$, the Gaussian weights and nodes are

$$w_j = 1, \tag{58}$$

$$x_j = c_j, \tag{59}$$

for $j = 1, \ldots, n$, whatever the functions ϕ_i are (since $\omega(0, x) = 0$, unless $x = c_j$ for some $j \in [1, n]$).

(iii) The quadrature weights and nodes depend continuously on α (by Corollary 4.2).

The intermediate problems which the continuation method solves are the Gaussian quadratures relative to the weight functions $\omega(\alpha, *)$. The scheme starts by setting $\alpha = 0$, then increases α in an adaptive manner until $\alpha = 1$, as follows. A current step size is maintained, by which α is incremented after each successful termination of Newton's method. After each unsuccessful termination of Newton's method, the step size is halved and the algorithm restarts from the point yielded by the last successful termination. After a certain number of successful steps, the current step size is doubled. (Experimentally, the current problem was found to be well suited to an aggressive mode of adaption: in the authors' implementation, the initial value of the step size was chosen to be one, and the step size was doubled after a single successful termination of Newton's method.)

5.1.1. Comparison to continuation method of [10]. The continuation method of this paper differs from the continuation method of [10] in that a different part of the system of equations is changed as a function of the continuation variable α. In [10], the thing changed is not the weight function ω but rather the functions $\phi_1, \ldots, \phi_{2n}$ which the quadrature is to integrate properly. Each of these functions is altered according to the formula

$$\phi_i(\alpha, x) = \alpha\psi_i(x) + (1 - \alpha)P_i(x), \tag{60}$$

where $\psi_1, \ldots, \psi_{2n}$ are the functions for which the quadrature was desired, and where $P_1, \ldots, P_n$ are some sequence of functions for which a Gaussian quadrature is known (for instance, polynomials). That continuation method has the drawback that the functions $\phi_1, \ldots, \phi_{2n}$ do not necessarily form an extended Chebyshev system when $0 < \alpha < 1$, even if the functions $\psi_1, \ldots, \psi_{2n}$ form an extended Chebyshev system. For instance, if the quadrature is to integrate two functions, $\psi_1 = P_2$, and $\psi_2 = P_1$, then when $\alpha = 1/2$, the functions ϕ_1 and ϕ_2 are identical, so the Jacobian matrix (33) is singular, whatever the (single) quadrature node x_1 might be.

5.1.2. Starting points. The choice of the points c_j was left indefinite above. In exact arithmetic the algorithm would converge for any choice of distinct points (see Lemma 4.1). However, the number of steps of the continuation method, and thus the speed of execution, is affected by the choice. More importantly, the numerical stability of the scheme might be compromised due to poor conditioning of the matrix J (see (33)). Indeed, while Lemma 4.1 guarantees that the matrix J is nonsingular, it says nothing about its condition number. Thus, in the authors' implementation, the points c_j used for the production of the quadrature of order n were computed from the nodes x_j of the quadrature of order $n - 1$ by the formulae

$$c_1 = x_1, \tag{61}$$

$$c_i = (x_{i-1} + x_i)/2, \qquad i = 2, \ldots, n - 1, \tag{62}$$

$$c_n = x_{n-1}. \tag{63}$$

With this choice, no failures to converge have been encountered in the authors' experience.

6. Numerical examples. A variety of quadratures were generated to illustrate the performance of the above algorithm. In each case the preprocessing step of producing singular functions was used. This step requires two sets of quadratures and interpolation schemes, which must approximate the desired kernel to the desired accuracy. These quadratures and interpolation schemes were chosen so that the approximation was accurate to about the precision of the arithmetic that was used. The following combination of quadrature and interpolation scheme which preserves inner products was used: the interval of integration was divided into several subintervals, and a combination of a (classical) Gaussian quadrature at Legendre nodes and polynomial interpolation was used on each subinterval.

In each of the following examples, the calculations were done in extended precision (Fortran `REAL*16`) arithmetic, with the exception of the last example, which was done in double precision (`REAL*8`) arithmetic.

6.1. Exponentials. In this example we construct quadratures for the integral

$$\int_0^\infty e^{-xt}dx \tag{64}$$

under the condition that $1 \le t \le 500$. In this case, the corresponding kernel $K : [0,\infty) \times [1,500] \to \mathbb{R}$ is given by

$$K(x,t) = e^{-xt} \tag{65}$$

and is ETP; thus its singular functions form an extended Chebyshev system. The measured maximum absolute error of integration of the produced quadratures, over the range $1 \le t \le 500$, is given, for selected n, in the following table.

n	6	8	14	23	27
Error	0.827E−03	0.726E−04	0.366E−07	0.356E−12	0.323E−14

The weights and nodes of the 27-point quadrature are included as Table 6.1; the remaining weights and nodes are available electronically at the URL `http://www.netlib.org/pdes/multipole/wts500.f`.

6.2. Complex exponentials. Here, we design quadratures for a new version [5] of the two-dimensional fast multipole method. These quadratures are for the integral

$$\int_0^\infty e^{-xz}dx, \tag{66}$$

under the condition that $z \in \mathbb{C}$ is constrained to lie in the region D of the complex plane which consists of the rectangle $[1,4]\times[-4,4]$ with a 1×1 square deleted from each of its two left-hand corners, as depicted in Figure 1. Since both the true integral (equal to $1/z$) and the quadrature which approximates the integral are complex analytic on that region, due to the maximum modulus principle the maximum error of the quadrature is achieved on the boundary δD of the region. Accordingly, the kernel whose singular functions were computed was $K(x,z) = e^{-xz}$, with z varying over δD. A brief examination of the resulting singular functions shows that they do not form a Chebyshev system; if they did so, the ith function would have $i-1$ zeros, yet it has many more. Thus, the algorithm is not guaranteed to work; however, it did so. The measured maximum absolute error of integration of the produced quadratures is given, for selected n, in the following table.

TABLE 6.1

27-point generalized Gaussian quadrature for decaying exponentials.

Node (x_i)	Weight (w_i)
0.5378759010624780E−03	0.1383311204046008E−02
0.2860176825815242E−02	0.3279869733166365E−02
0.7148658617716300E−02	0.5330932895600203E−02
0.1360965515937845E−01	0.7646093110803760E−02
0.2257800188133212E−01	0.1037458793227033E−01
0.3456421989535069E−01	0.1372178039022047E−01
0.5032042618508775E−01	0.1796868836009351E−01
0.7092509447124836E−01	0.2348971809947674E−01
0.9788439120828463E−01	0.3076860552710760E−01
0.1332509921950535E+00	0.4041894092839717E−01
0.1797695570864978E+00	0.5321827718681367E−01
0.2410654714132133E+00	0.7016094768858448E−01
0.3218961915636380E+00	0.9253048536912244E−01
0.4284852078938826E+00	0.1219928996130354E+00
0.5689615509235298E+00	0.1607156476580828E+00
0.7539347736933301E+00	0.2115215602167892E+00
0.9972472224438443E+00	0.2780925850550500E+00
0.1316964566299846E+01	0.3652478333806065E+00
0.1736698582009859E+01	0.4793398853949993E+00
0.2287418444638146E+01	0.6288554258416082E+00
0.3010034073439038E+01	0.8254021100491956E+00
0.3959315495048493E+01	0.1085495633209734E+01
0.5210381702393131E+01	0.1434174907278760E+01
0.6870768194824406E+01	0.1913323186889750E+01
0.9106577764323245E+01	0.2604342790201154E+01
0.1221294512896673E+02	0.3708436699287805E+01
0.1689348652665484E+02	0.6023086156615004E+01

n	7	10	17	26	32
Error	0.107E−02	0.398E−04	0.156E−07	0.801E−12	0.282E−14

The weights and nodes of the quadratures are available electronically at the URL `http://www.netlib.org/pdes/multipole/pwts4.f`.

6.3. Exponentials multiplied by I_0. In this example, quadrature formulae are constructed for integrals of the form

$$\int_0^\infty I_0(xy)e^{-xt}dx, \tag{67}$$

under the condition that $t \in [1, 500]$ and $y \in [0, t-1]$; these formulae were designed to be used in a version of the one-dimensional fast multipole method which is used in an algorithm [6] for the fast Hankel transform. In this case the singular functions produced by the precomputation stage were extremely similar to those for exponentials alone; unlike in the case of complex exponentials, it is possible that they form a Chebyshev system. In any case, the algorithm converged, producing a quadrature which required two more nodes for double precision accuracy than were required for the integration of exponentials alone. The measured maximum absolute error of integration of the produced quadratures is given, for selected n, in the following table.

n	6	8	14	24	29
Error	0.997E−03	0.892E−04	0.900E−07	0.925E−12	0.299E−14

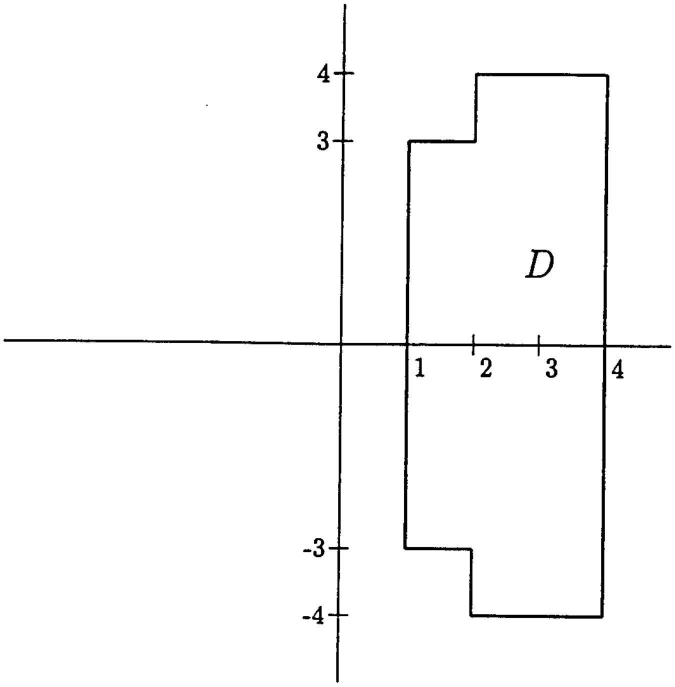


FIG. 1. *Range of coefficient z of complex exponentials to be integrated.*

The weights and nodes of the quadratures are available electronically at the URL `http://www.netlib.org/pdes/multipole/swts500.f`.

6.4. Exponentials multiplied by J_0. Here, we construct quadratures for the integral

$$\int_0^\infty J_0(xy)e^{-xt}dx, \tag{68}$$

under the conditions that $t \in [1,4]$ and $y \in [0, 4\sqrt{2}]$, and where J_0 denotes the Bessel function of the first kind of order zero. These quadratures are used in a new version [4] of the three-dimensional fast multipole method. J_0 is given by the well-known (see for instance [1]) formula

$$J_0(y) = \frac{1}{\pi}\int_0^\pi \cos(y\cos\theta)d\theta. \tag{69}$$

Substituting (69) into (68) yields the integral

$$\begin{aligned} &\int_0^\infty \left(\frac{1}{\pi}\int_0^\pi \cos(xy\cos\theta)d\theta\right) e^{-xt}dx \\ &= \frac{1}{\pi}\int_0^\pi \int_0^\infty \cos(xy\cos\theta)e^{-xt}dxd\theta. \end{aligned} \tag{70}$$

Thus a quadrature accurate for the integral

$$\int_0^\infty \cos(xy)e^{-xt}dx, \tag{71}$$

under the conditions that $t \in [1,4]$ and $y \in [0, 4\sqrt{2}]$, is also accurate for the integral (68) under the same conditions on y and t. Since the function $\cos(xy)e^{-xt}$ is a harmonic function of y and t, by the maximum modulus principle the maximum error of a quadrature for (71) lies on the boundary δD of the rectangular region $t \in [1,4]$, $y \in [0, 4\sqrt{2}]$. Accordingly, the kernel whose singular functions were computed was $K(x,z) = \cos(xy)e^{-xt}$, with (t,y) varying over δD. As in the case of complex exponentials, the singular functions have too many zeros to form a Chebyshev system; however, the algorithm converged.

The measured maximum absolute error of integration of the produced quadratures is given, for selected n, in the following table.

n	8	12	21	31	40
Error	0.162E−02	0.709E−04	0.553E−07	0.195E−10	0.147E−13

The weights and nodes of the quadratures are available electronically at the URL `http://www.netlib.org/pdes/multipole/vwts.f`.

6.5. Numerical observations. The following observations were made in the course of our numerical experiments.

(i) The number of continuation steps required is highly variable; in many cases, only one step sufficed to produce the quadrature; less frequently, up to fifty or so continuation steps were required. This variability occurred even between quadratures for successive numbers n of nodes, with the same weight function and kernel K.

(ii) The algorithm worked in the cases where Theorem 2.1 applied, and also in cases where it did not. In the latter cases, it is conceivable that the resulting quadratures would have negative weights or that they would not be unique. However, all computed weights were positive, and, while no systematic attempt was made to look for nonuniqueness of the quadratures, no instance of it was observed.

7. Generalizations and applications.

(i) The success of the algorithm in instances where Theorem 2.1 does not apply suggests that further theoretical investigation of conditions for the existence of generalized Gaussian quadratures would be profitable.

(ii) An obvious generalization of these results is to quadratures for integrals in more than one dimension. However, such an extension does not seem to have been explored classically; the authors are investigating a generalization of Theorem 2.1 for multidimensional quadratures.

(iii) An obvious application of the algorithm of this paper is for the efficient evaluation of functions represented by their integral transforms (see sections 6.1, 6.2, 6.3, 6.4 above, as well as [5] and [4]). The method of steepest descent in the numerical complex analysis provides a wide field of applications for such algorithms.

(iv) An entirely different field of applications involves the numerical solution of integral equations with singular kernels; of particular interest are boundary integral equations of scattering theory on regions with corners. The authors are currently pursuing this direction of research.

REFERENCES

[1] M. Abramowitz and I. Stegun, *Handbook of Mathematical Functions*, Applied Mathematics Series, National Bureau of Standards, Washington, DC, 1964.

[2] F. GANTMACHER AND M. KREIN, *Oscillation Matrices and Kernels and Small Oscillations of Mechanical Systems*, 2nd ed., Gosudarstv. Izdat. Tehn-Teor. Lit., Moscow, 1950 (in Russian).

[3] G. H. GOLUB AND C. H. VAN LOAN, *Matrix Computations*, Johns Hopkins University Press, Baltimore, 1983.

[4] L. GREENGARD AND V. ROKHLIN, *A new version of the fast multipole method for the Laplace equation in three dimensions*, Acta Numerica, 6 (1997), pp. 229–269.

[5] T. HRYCAK AND V. ROKHLIN, *An Improved Fast Multipole Algorithm for Potential Fields*, Research Report 1089, Computer Science Department, Yale University, New Haven, CT, 1995.

[6] S. KAPUR AND V. ROKHLIN, *An Algorithm for the Fast Hankel Transform*, Technical Report 1045, Computer Science Department, Yale University, New Haven, CT, 1995.

[7] S. KARLIN, *The existence of eigenvalues for integral operators*, Trans. Amer. Math. Soc., 113 (1964), pp. 1–17.

[8] S. KARLIN AND W. J. STUDDEN, *Tchebycheff Systems with Applications in Analysis and Statistics*, John Wiley (Interscience), New York, 1966.

[9] M. G. KREIN, *The Ideas of P. L. Chebyshev and A. A. Markov in the Theory of Limiting Values of Integrals*, Amer. Math. Soc. Transl. 2, AMS, Providence, RI, 1959, pp. 1–122.

[10] J. MA, V. ROKHLIN AND S. WANDZURA, *Generalized Gaussian quadrature rules for systems of arbitrary functions*, SIAM J. Numer. Anal., 34 (1996), pp. 971–996.

[11] A. A. MARKOV, *On the limiting values of integrals in connection with interpolation*, Zap. Imp. Akad. Nauk. Fiz.-Mat. Otd. (8) 6 (1898), no. 5 (in Russian); pp. 146–230 of [12].

[12] A. A. MARKOV, *Selected Papers on Continued Fractions and the Theory of Functions Deviating Least from Zero*, OGIZ, Moscow-Leningrad, 1948 (in Russian).

[13] M. REED AND B. SIMON, *Methods of Modern Mathematical Physics*, Vol. 1, Academic Press, New York, 1980.

[14] J. STOER AND R. BULIRSCH, *Introduction to Numerical Analysis*, 2nd ed., Springer-Verlag, New York, 1993.

To appear in JCP

AN INTEGRAL EVOLUTION FORMULA FOR THE WAVE EQUATION*

BRADLEY ALPERT[†], LESLIE GREENGARD[‡], AND THOMAS HAGSTROM[§]

Abstract. We present a new time-symmetric evolution formula for the scalar wave equation. It is simply related to the classical D'Alembert or spherical means representations, but applies equally well in two space dimensions. It can be used to develop stable, robust numerical schemes on irregular meshes.

1. Introduction.

It is notoriously difficult to construct stable high-order explicit marching schemes for the wave equation on irregular meshes. The time-step restriction is typically determined by the smallest cell present in the discretization. In this note, we describe a new approach to the construction of stable, explicit schemes, based on a simple time-symmetric evolution formula.

Initially we consider the Cauchy problem in $\mathbf{R}^d$,

$$\begin{aligned} u_{tt} &= \Delta u, \\ u(\mathbf{x},0) &= u_0(\mathbf{x}), \\ u_t(\mathbf{x},0) &= v_0(\mathbf{x}), \end{aligned} \tag{1.1}$$

where Δ denotes the Laplacian operator. In one space dimension, the solution can be written using D'Alembert's formula as

$$u(x,t) = \frac{1}{2}(u_0(x-t) + u_0(x+t)) + \int_{x-t}^{x+t} v_0(s)\,ds. \tag{1.2}$$

We can eliminate the term involving the data $v_0(x)$ by using the time-symmetric form:

$$u(x,t) + u(x,-t) = u(x-t,0) + u(x+t,0). \tag{1.3}$$

In three dimensions, the analog of (1.3) is the spherical means formula [2, 4, 5]

$$u(\mathbf{x},t) + u(\mathbf{x},-t) = \frac{\partial}{\partial t}\left[\frac{t}{4\pi}\int_{|\mathbf{y}-\mathbf{x}|=t} u(\mathbf{y},0)d\sigma\right], \tag{1.4}$$

where $d\sigma$ is an element of surface area. In two dimensions, the situation is slightly more complex because of the absence of a strong Huygen's principle. The solution

---

[*] Contribution of U.S. government not subject to copyright. Key words: small-cell problem, stability. Subject classification: 35C15, 35L05, 65M12

[†] National Institute of Standards and Technology, 325 Broadway, Boulder, CO 80303. *email:* alpert@boulder.nist.gov The work of this author was supported in part by the DARPA Applied and Computational Mathematics Program under appropriation 9770400.

[‡] Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY 10012-1110. *email:* greengar@cims.nyu.edu The work of this author was supported in part by the U.S. Department of Energy under contract DEFGO288ER25053 and DARPA/AFOSR under contract F94620-95-C-0075.

[§] Department of Mathematics and Statistics, University of New Mexico, Albuquerque, NM 87131. *email:* hagstrom@math.unm.edu The work of this author was supported in part by NSF Grant DMS-9600146, DARPA/AFOSR Contract F94620-95-C-0075, and, while in residence at the Courant Institute, DOE Contract DEFGO288ER25053.

depends not just on function values over the boundary of the disk of radius t, but on all values in its interior:

$$u(\mathbf{x},t)+u(\mathbf{x},-t)=\frac{\partial}{\partial t}\left[\frac{1}{2\pi}\int_{|\mathbf{y}-\mathbf{x}|\le t}\frac{u(\mathbf{y},0)}{\sqrt{t^2-|\mathbf{x}-\mathbf{y}|^2}}d\mathbf{y}\right]. \tag{1.5}$$

For numerical computation, formulas of the type (1.3), (1.4), and (1.5) are not widely used because they do not suggest a procedure at physical boundaries and are not easily extended to more general partial differential equations.

2. A central difference evolution formula. Consider the Fourier transform of the wave function $u(\mathbf{x},t)$, namely

$$U(\mathbf{k},t)=\left(\frac{1}{\sqrt{2\pi}}\right)^d\int_{\mathbf{R}^d}e^{-i\mathbf{k}\cdot\mathbf{x}}u(\mathbf{x},t)d\mathbf{x}.$$

The partial differential equation in (1.1) can then be replaced by

$$U_{tt}(\mathbf{k},t)=-|\mathbf{k}|^2U(\mathbf{k},t).$$

Solving this ordinary differential equation, we obtain

$$U(\mathbf{k},t)+U(\mathbf{k},-t)=2U(\mathbf{k},0)\cos(|\mathbf{k}|t)$$

or

$$U(\mathbf{k},t)-2U(\mathbf{k},0)+U(\mathbf{k},-t)=\left[\frac{2\cos(|\mathbf{k}|t)-2}{-|\mathbf{k}|^2}\right](-|\mathbf{k}|^2)\,U(\mathbf{k},0). \tag{2.1}$$

Our main result follows.

THEOREM 2.1. *Let $u(\mathbf{x},t)$ denote a solution to the homogeneous wave equation*

$$u_{tt}=\Delta u$$

in $\mathbf{R}^d$. Then

$$u(\mathbf{x},t)-2u(\mathbf{x},0)+u(\mathbf{x},-t)=\int_{|\mathbf{y}-\mathbf{x}|\le t}G_d(|\mathbf{x}-\mathbf{y}|,t)\,\Delta u(\mathbf{y},0)\,d\mathbf{y}, \tag{2.2}$$

where

$$G_1(r,t)=t-r \tag{2.3}$$

$$G_2(r,t)=\ln(t+\sqrt{t^2-r^2})-\ln r \tag{2.4}$$

$$G_3(r,t)=\frac{1}{r} \tag{2.5}$$

Proof. The formula (2.2) is obtained from the convolution theorem by transforming (2.1) back to physical space. We provide a few more details for two space dimensions, where we need to evaluate the kernel

$$G_2(|\mathbf{x}|,t)=\frac{1}{2\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left[\frac{2\cos(|\mathbf{k}|t)-2}{-|\mathbf{k}|^2}\right]\cdot e^{i\mathbf{k}\cdot\mathbf{x}}\,d\mathbf{k}.$$

Changing to polar coordinates, we have

$$G_2(r,t) = \frac{1}{2\pi}\int_0^\infty \int_0^{2\pi} \left[\frac{2-2\cos(kt)}{k^2}\right] e^{ikr\cos(\theta-\phi)}\, k\, dk\, d\phi$$
$$= \int_0^\infty \left[\frac{2-2\cos(kt)}{k}\right] J_0(kr)\, dk,$$

where $\mathbf{k} = (k\cos\phi, k\sin\phi)$, $\mathbf{x} = (r\cos\theta, r\sin\theta)$, and J_0 denotes the Bessel function of order zero. The desired result now follows from the formula ([1], 6.693)

$$\int_0^\infty J_\nu(kr)\cos(kt)\frac{dk}{k} = \begin{cases} \frac{1}{\nu}\cos(\nu \arcsin\frac{t}{r}) & t \le r \\ \frac{r^\nu}{\nu(t+\sqrt{t^2-r^2})^\nu}\cos\frac{\nu\pi}{2} & t \ge r, \end{cases}$$

with some care in taking the limit $\nu \to 0$. □

REMARK 2.2. Integration by parts and Green's identities can be used to recover the formulas (1.3), (1.4), and (1.5) from (2.2).

REMARK 2.3. Our evolution scheme can be viewed as an integral form of the widely-used Lax-Wendroff method. The latter method uses central differencing in time to generate the series

$$u(\mathbf{x},t) - 2u(\mathbf{x},0) + u(\mathbf{x},-t) = t^2 u_{tt}(\mathbf{x},0) + \frac{t^4}{12}u_{tttt}(\mathbf{x},0) + \frac{t^6}{360}u_{tttttt}(\mathbf{x},0) + \cdots.$$

Replacing the time derivatives with powers of the Laplacian, one obtains

$$u(\mathbf{x},t) - 2u(\mathbf{x},0) + u(\mathbf{x},-t) = t^2 \Delta u(\mathbf{x},0) + \frac{t^4}{12}\Delta^2 u(\mathbf{x},0) + \frac{t^6}{360}\Delta^3 u(\mathbf{x},0) + \cdots.$$

Once a numerical approximation is chosen for the Laplacian operator, the Lax-Wendroff scheme achieves arbitrary order accuracy in time by incorporating higher and higher powers of the Laplacian in a three time level scheme. Stability and spatial accuracy depend, of course, on how the Laplacian is computed.

3. Forcing.

We next consider the wave equation with a source term

$$u_{tt} = \Delta u + f \tag{3.1}$$

which from Fourier transformation ($u \to U$, $f \to F$) becomes

$$U_{tt}(\mathbf{k},t) = -|\mathbf{k}|^2 U(\mathbf{k},t) + F(\mathbf{k},t),$$

whose solution is given by

$$U(\mathbf{k},t) - 2U(\mathbf{k},0) + U(\mathbf{k},-t) = 2\big[\cos(|\mathbf{k}|t) - 1\big]U(\mathbf{k},0) + \int_{-t}^{t} \frac{\sin\left(|\mathbf{k}|(t-|s|)\right)}{|\mathbf{k}|} F(\mathbf{k},s)\, ds.$$

The identity

$$\frac{\sin(|\mathbf{k}|t)}{|\mathbf{k}|} = -\frac{\partial}{\partial t}\left(\frac{\cos(|\mathbf{k}|t) - 1}{|\mathbf{k}|^2}\right)$$

and integration by parts, in combination with (2.2), now yield

THEOREM 3.1. *Let $u(\mathbf{x},t)$ denote a solution to the inhomogeneous wave equation* (3.1) *in $\mathbf{R}^d$. Then*

$$u(\mathbf{x},t) - 2u(\mathbf{x},0) + u(\mathbf{x},-t) = \int_{|\mathbf{y}-\mathbf{x}|\le t} G_d(|\mathbf{x}-\mathbf{y}|,t)\big[\Delta u(\mathbf{y},0) + f(\mathbf{y},0)\big]\,d\mathbf{y} \tag{3.2}$$

$$+\frac{1}{2}\int_{-t}^{t} \text{signum}(s) \int_{|\mathbf{y}-\mathbf{x}|\le t-|s|} G_d(|\mathbf{x}-\mathbf{y}|,t-|s|)\,f'(\mathbf{y},s)\,d\mathbf{y}\,ds,$$

where G_d is given in (2.3)–(2.5) *and $f'(\mathbf{x},t) = \partial f(\mathbf{x},t)/\partial t$.*

REMARK 3.2. The derivative f' of the forcing term may be analytically removed from (3.2) by integration, yielding formulas that differ somewhat for $d = 1,2,3$. In three dimensions, for example, the double integral reduces to the particularly simple form

$$\frac{1}{2}\int_{|\mathbf{y}-\mathbf{x}|\le t} \frac{f(\mathbf{y},|\mathbf{x}-\mathbf{y}|-t) - 2f(\mathbf{y},0) + f(\mathbf{y},t-|\mathbf{x}-\mathbf{y}|)}{|\mathbf{x}-\mathbf{y}|}\,d\mathbf{y}.$$

4. Discretization.

In order to use formula (2.2) or (3.2) for computation, we need to evaluate the integral

$$Q_u(\mathbf{x}) = \int_{|\mathbf{y}-\mathbf{x}|\le t} G_d(|\mathbf{x}-\mathbf{y}|,t)\Delta u(\mathbf{y},0)\,d\mathbf{y}, \tag{4.1}$$

for each discretization point $\mathbf{x}$. In this brief note, we will restrict our attention to the one-dimensional case. Away from physical boundaries, there are three clear options:

1. Use a quadrature formula designed for formula (4.1):

$$Q_u(x) = \int_{x-t}^{x+t} (t - |y-x|)u_{yy}(y,0)\,dy. \tag{4.2}$$

2. Integrate by parts once to obtain

$$Q_u(x) = -\int_{x-t}^{x} u_y(y,0)\,dy + \int_{x}^{x+t} u_y(y,0)\,dy. \tag{4.3}$$

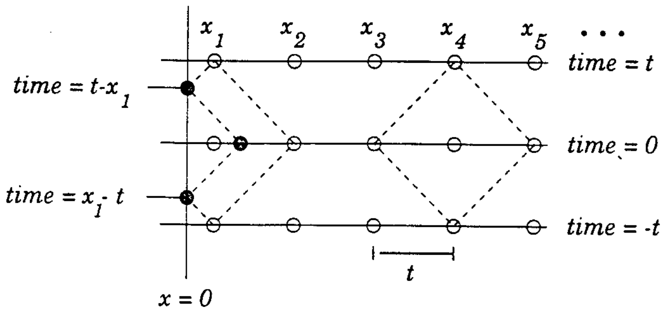
3. Integrate by parts twice to obtain

$$Q_u(x) = u(x-t,0) - 2u(x,0) + u(x+t,0).$$

All three formulas are exact (the last yielding the time-symmetric scheme (1.3)). In the first case, one needs to approximate u_{xx} within the domain of dependence. In the second case, one needs to approximate u_x within the domain of dependence. In the third case, one needs to interpolate $u(x-t,0)$ and $u(x+t,0)$ from the possibly irregular mesh points where $u(x,0)$ is known. The stability of each scheme will depend on how the interpolation/approximation problem is handled.

To demonstrate the value of the integral formulation, we suppose that we are solving the problem (1.1) with the Dirichlet boundary condition $u(0,t) = g(t)$. For the sake of simplicity, we assume that the grid spacing in x is equal to the time step t. The only irregular point is the first grid point x_1 which is arbitrarily close to the boundary $x = 0$, creating what is often referred to as a *small cell* problem (Fig. 4.1).

FIG. 4.1. *An irregular mesh in one space dimension. The grid points $x_1, x_2, x_3, \dots$ are equispaced, but the first grid point is near the physical boundary $x = 0$. At regular grid points, the symmetric stencil (1.3) is used. For the node x_1, the interpolatory scheme described in section 4.2 uses the indicated stencil. It requires values at the irregular points marked by darkened circles.*



For nodes other than x_1, we can use any of the three options outlined above. For $x \le x_1 \le t$, let us define

$$\tilde{u}(x,\tau) = 2u(x,0) - u(x,-\tau) + \int_0^{x+\tau} (\tau - |y-x|)\, u_{yy}(y,0)\, dy. \tag{4.4}$$

Note that $\tilde{u}$ satisfies the wave equation exactly, under the assumption that the function $u_{xx}(x,0)$ is extended outside the domain $x \ge 0$ by zero. Taking into account the Dirichlet data, it is straightforward to verify that the exact solution is

$$u(x_1,t) = \tilde{u}(x_1,t) + g(t-x_1) - \tilde{u}(0,t-x_1). \tag{4.5}$$

4.1. Quadrature schemes. The most straightforward use of the quadrature approach is to compute u_{xx} at time $t = 0$ by a finite difference method of kth order accuracy. We can then integrate the formula (4.2) or (4.4) exactly for a polynomial approximant of u_{xx} of degree $k-1$. For $k = 2$ this involves computing the second derivative using the usual 3-point stencil at regular grid points and a one-sided 4-point stencil for the irregular points $x = 0, x_1$. The necessary quadratures are easy to derive for a piecewise linear approximation of u_{xx}.

4.2. Interpolation schemes. Integrating by parts yet again, we can rewrite the formula (4.4) for $\tilde{u}(x_1,t)$ as

$$\tilde{u}(x_1,t) = -u(x,-t) + u(x_1+t,0) + u(0,0) - (t-x_1)u_x(0,0).$$

Combining this result with (4.5), we have

$$\begin{aligned} u(x_1,t) = {} & -u(x_1,-t) + u(x_1+t,0) \\ & +g(t-x_1) + g(-t+x_1) + u(t-x_1,0). \end{aligned} \tag{4.6}$$

For regular grid points, we use the exact formula (1.3). Once we choose a method for approximating the values $g(t-x_1)$, $g(-t+x_1)$, and $u(t-x_1,0)$, we have a well-defined evolution scheme. In our numerical experiments, we assume the Dirichlet data $g(t)$ is known analytically, so that we only need to interpolate $u(t-x_1,0)$.

4.3. Extrapolation schemes. As a final alternative, one can try to use the time symmetric formula (1.3) for all grid points. This involves the value $u(x_1-t,0)$, which requires extrapolation from the known data at $x = 0, x_1, x_2, \dots$

TABLE 5.1

Performance of the quadrature, interpolation, extrapolation, and leapfrog schemes. The first column lists the number of subintervals in the uniform grid region. The second through the fifth columns list the L_2 error from using the indicated evolution scheme after N steps.

N	$E_2(Q2)$	$E_2(I1)$	$E_2(I3)$	$E_2(X1)$	$E_2(LF2)$
16	$0.58 \cdot 10^{-4}$	$0.31 \cdot 10^{-6}$	$0.79 \cdot 10^{-10}$	$0.17 \cdot 10^{54}$	$0.35 \cdot 10^{75}$
32	$0.12 \cdot 10^{-4}$	$0.16 \cdot 10^{-6}$	$0.97 \cdot 10^{-11}$	$0.39 \cdot 10^{104}$	$0.42 \cdot 10^{150}$
64	$0.28 \cdot 10^{-5}$	$0.81 \cdot 10^{-7}$	$0.11 \cdot 10^{-11}$	–	–
128	$0.67 \cdot 10^{-6}$	$0.41 \cdot 10^{-7}$	$0.14 \cdot 10^{-12}$	–	–
256	$0.16 \cdot 10^{-6}$	$0.19 \cdot 10^{-7}$	$0.27 \cdot 10^{-13}$	–	–

5. A numerical example.

We have implemented simple versions of the various methods described above: the second order quadrature scheme ($Q2$), the interpolation scheme using linear and cubic interpolation ($I1$, $I3$), and the extrapolation scheme using linear approximation ($X1$). For the sake of comparison, we use the same values of u_{xx} as in the quadrature approach, but march using the simplest leapfrog scheme [3]

$$\tilde{u}(x,t) = 2u(x,0) - u(x,-t) + t^2 u_{xx}(x,0). \tag{5.1}$$

We will denote this method by $LF2$.

We consider the wave equation on $[0,1]$ as an initial/boundary-value problem with exact solution $\sin(x-t) + \sin(x - t - \frac{1}{2})$. We set $x_1 = 1.0 \cdot 10^{-5}$, $x_{N+1} = 1 - 1.0 \cdot 10^{-6}$, and place $N-1$ equispaced points on the interval $[x_1, x_{N+1}]$. With $N = 16, 32, 64, 128, 256$, both the first and last cells are extremely small in comparison with $\Delta t = (x_{N+1} - x_1)/N$. The calculation is terminated after N steps, at which point we measure the L_2 error of the solution. The scheme used at the right boundary ($x = 1$) is analogous to the one described above at the left boundary ($x = 0$).

Results of the methods $Q2$, $I1$, $I3$, $X1$, $LF2$ are summarized in Table 5.1.

$Q2$, $I1$, and $I3$ appear to be stable, while both the extrapolation and leapfrog schemes diverge. It is also worth noting that $Q2$ is globally second order accurate, $I1$ is globally first order accurate, and $I3$ is globally third order accurate. This is consistent with a straightforward local error analysis. The reason that the first order scheme $I1$ is more accurate than $Q2$ for small N is that we are using an exact formula away from the irregular nodes in the former and a second order accurate quadrature at all points in the latter.

6. Conclusions.

We have derived a new exact representation for solutions of the wave equation. Theorem 2.1 and theorem 3.1 may be of analytical interest in their own right, but we have concentrated in this note on exploring some numerical consequences. We believe that marching schemes based on this approach have advantageous stability properties when compared to existing methods, most notably in removing the "small cell" problems which arise when using unstructured grids or regular Cartesian meshes in complex geometries. Although small cells can be easily eliminated in one dimension, at some cost in accuracy, doing so in two or three dimensions is more complicated and results in greater loss of accuracy. Furthermore, higher-order discretizations *require* small cells near the boundary to avoid the Runge phenomenon.

We have illustrated the advantages in the simplest one-dimensional model problem, but the extension to higher dimensions is straightforward. Suppose, for example, that we are solving the wave equation in a domain $\Omega \subset \mathbf{R}^d$. If a point $\mathbf{x}$ is within a time step t of the domain boundary $\partial\Omega$, we define the function

$$\tilde{u}(\mathbf{x},\tau) = 2u(\mathbf{x},0) - u(\mathbf{x},-\tau) + \int_{S_\tau \cap \Omega} G_d(|\mathbf{x}-\mathbf{y}|,\tau)\,\Delta u(\mathbf{y},0)\,d\mathbf{y} \tag{6.1}$$

where $S_\tau = \{\mathbf{y} : |\mathbf{y}-\mathbf{x}| \leq \tau\}$. Whereas in one dimension, the exact solution is given by (4.5), it is now of the form

$$u(\mathbf{x},t) = \tilde{u}(\mathbf{x},t) + B(\partial\Omega,\tilde{u},g). \tag{6.2}$$

The operator $B(\partial\Omega,\tilde{u},g)$ describes the exact solution to the Dirichlet problem with zero initial data and boundary condition $g(\mathbf{x},t) - \tilde{u}(\mathbf{x},t)$. This can be written out explicitly in terms of hyperbolic potential theory and can easily be generalized to Neumann or Robin boundary value problems.

It is not surprising, perhaps, that robustness and stability come at a price. In our formulation, that price is the construction of appropriate quadratures for both the volume integral in (6.1) and the boundary operator $B(\partial\Omega,\tilde{u},g)$ in (6.2). Higher dimensional examples, higher-order discretizations, and stability estimates will be reported at a later date.

REFERENCES

[1] I. S. GRADSHTEYN AND I. M. RYZHIK (1980), *Tables of Integrals, Series, and Products*, Academic Press, New York.

[2] P. R. GARABEDIAN (1964), *Partial Differential Equations*, Wiley, New York.

[3] ARIEH ISERLES (1996), *A First Course in the Numerical Analysis of Differential Equations*, Cambridge University Press.

[4] F. JOHN (1982), *Partial Differential Equations*, Springer-Verlag, New York.

[5] F. JOHN (1955), *Plane Waves and Spherical Means Applied to Partial Differential Equations*, Interscience Publishers, New York.

Quadrature Rules on Triangles in R^2

Stephen Wandzura[†] and Hong Xiao[‡]

Research Report YALEU/DCS/RR-1168
November 30, 1998

YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

We present quadrature rules on triangles in R^2, somewhat similar to Gaussian rules on intervals in R^1. By a scheme combining simple group theory and numerical optimization, we obtain rules of orders up to 30, which is more than twice the order of previously available rules of similar efficiency. Also discussed are generalizations of our scheme to other regions in R^2, and to higher dimensions.

Quadrature Rules on Triangles in R^2

Stephen Wandzura[†] and Hong Xiao[‡]
Research Report YALEU/DCS/RR-1168
November 30, 1998

[†] HRL Laboratories, 3011 Malibu Canyon Road, Malibu, CA 90265. The work of this author was supported in part by the Defense Advanced Research Projects Agency (DARPA) of the U.S. Department of Defense under Air Force Office of Scientific Research Contract No. F49620-91-C-0064, and in part by DARPA Contract No. MDA972-95-C-0021.

[‡] Department of Computer Science, Yale University, New Haven, CT 06520. The work of this author was supported by DARPA/AFOSR under Grant F49620-97-1-0011.

Approved for public release: distribution is unlimited.

Keywords: *Quadratures, Triangles, Polynomials.*

1 Introduction

Gaussian quadratures are a classical tool of numerical integration and possess several desirable features such as uniqueness of nodes, positivity of weights, and an optimal number of nodes: an N-point Gaussian rule is exact for all polynomials of orders up to $2N-1$, and no N-point rule is exact for all polynomials of order $2N$. Since many situations require high-order quadratures in dimensions greater than one, a number of attempts have been made to construct quadrature rules in two dimensions that resemble the Gaussian ones. Of particular interest are quadrature rules on triangles, which are a standard tool for describing surfaces, and in many other situations.

One widely used approach is the naive tensor product rule, based on one dimensional quadratures. This approach is effective when the region of integration is a parallelogram. It is fairly straightforward to construct "tensor product" quadrature rules on triangles (see, for example, [12]) and on certain other polygons. However, the resulting quadrature rules are less efficient than those on rectangles. Furthermore, tensor product rules lack symmetry on triangles, a convenient feature for programming.

Lyness and Jespersen performed an exhaustive study of quadrature rules on triangles, and developed two types of fairly efficient rules which they termed "holistic" and "cytolic". They generated rules of orders up to twelve [6]. Berntsen and Espelid constructed quadrature rules of degree 13 for the triangle [1].

We present a scheme for the generation of reasonably high-order quadratures for polynomials on triangles in R^2. The scheme is based on the simple observation that integrals over regularly shaped regions are invariant under certain transformations. It is essentially a formalization and generalization of the approach used in [6]. With this scheme, quadrature rules of orders up to 30 on triangles have been obtained.

The structure of this paper is as follows. In Section 2 we introduce mathematical and numerical preliminaries. In Section 3 we develop the analytical apparatus used in the construction of the quadrature rules. We describe our scheme in Section 4, and illustrate it with the construction of quadratures on a standard triangle in Section 5. Finally Section 6 contains discussions and conclusions.

2 Mathematical and Numerical Preliminaries

In this section, we collect the relevant mathematical and numerical tools to be used in Section 3.

2.1 Representation Theory

Following is a summary of several elementary facts about representations of finite groups; a more detailed discussion on this subject can be found, for example, in [13].

Suppose that Q is a region in the xy-plane and G the symmetry group of Q. As is well-known from the theory of representations, the points of Q may be partitioned into G-orbits, each of which is an equivalence class on Q with respect to the relation defined by the group G.

Similarly, the function spaces on Q may be partitioned into subspaces, each containing functions that transform according to a particular irreducible representation (IR) of G. Furthermore, the inner product, defined as

$$(f_i, f_j) \equiv \iint_Q f_i(x,y) \cdot f_j(x,y)\; dx\, dy\,, \tag{1}$$

vanishes if f_i and f_j transform according to distinct IRs.

Two immediate consequences follow from the preceding fact:

- Since
$$\iint_Q f(x,y)\, dx\; dy = (1, f)\,, \tag{2}$$
any function not belonging to the identity representation of G integrates to zero.
- If the set $\{(x_i, y_i)\}$, $i = 1, 2, \ldots, n$, constitute a G-orbit in Q, then the function
$$\sum_{i=1}^{n} \delta(x - x_i, y - y_i)$$
belongs to the identity representation; thus
$$\sum_{i=1}^{n} f(x_i, y_i) = 0$$
for any f belonging to any IR other than the identity.

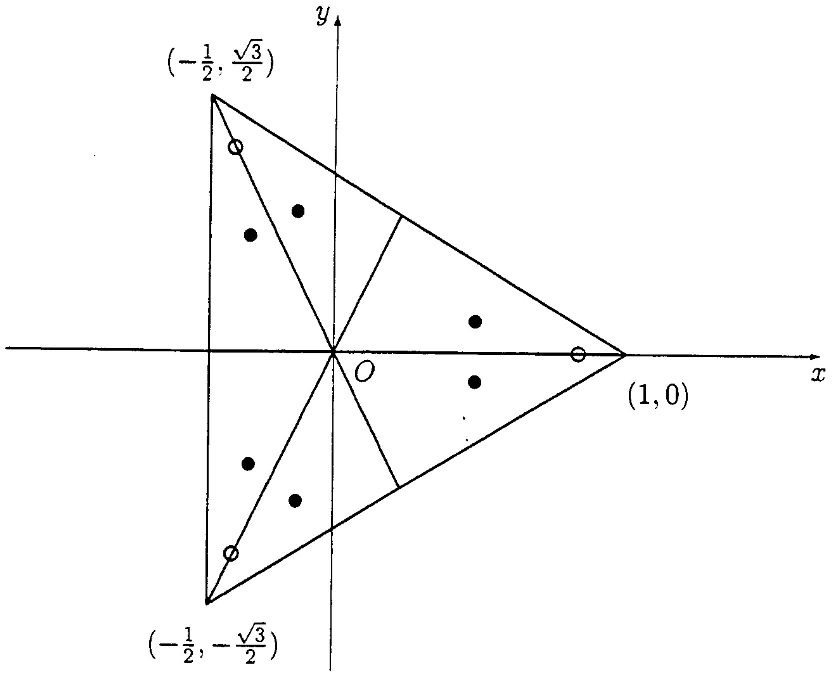
In other words, when constructing a quadrature rule that is invariant under G, we need only adjust the weights and abscissae to correctly integrate functions belonging to the identity representation; all functions belonging to nontrivial representations are integrated exactly.

Conveniently, the operator that projects onto the function subspace transforming according to the identity representation is given by a sum over transformed functions:

$$(P_E f)(x,y) = \frac{1}{m} \sum_{l=1}^{m} \Phi_{g_l}(f)(x,y)\,, \tag{3}$$

where $\Phi_{g_l}(f)$ denotes that transformation of $f(x,y)$ according to $g_l \in G$, and m, the order of G.

We will denote the standard equilateral triangle in R^2 by T (see Figure 2.1). In this case, the symmetry group is usually denoted by D_3, and is of order 6. The points of T are naturally classified by D_3-orbits, each orbit containing one, three, or six points. The first class consists of the center of the triangle; the second class consists of the union of three medians minus the center of the triangle; the third class consists of all points on T not belonging to the first and second classes.



y
$(-\frac{1}{2}, \frac{\sqrt{3}}{2})$
O
x
$(1,0)$
$(-\frac{1}{2}, -\frac{\sqrt{3}}{2})$

Figure 2.1. An Equilateral Triangle

2.2 Simulated Annealing

Simulated annealing is a numerical technique for solving combinatorial optimization problems, originally developed by Kirkpatrick et al [4, 5]. The algorithm draws an analogy between the behavior of a physical system with many degrees of freedom in thermal equilibrium at a series of finite temperatures as encountered in statistical physics, and the problem of finding the minimum of a given function of many parameters as in combinatorial optimization. It was based on a simple idea [3]:

> When optimizing a very large and complex system (i.e., a system with many degrees of freedom), instead of "always" going downhill, try to go downhill "most of the time."

There are three main components in any application of simulated annealing method; they are:

- Configure the optimization problem into a many-body physical system with states S.
- Construct a scalar objective function $E(S)$, which corresponds to the energy function of a physical system; the optimization problem then becomes finding the minimum energy configuration of the physical system.
- Construct a system of random state modifications (or updates) that obey detailed balance [8]; the random state modifications must ensure that any allowable state of the system is reachable.

- Develop the annealing (or cooling) schedule which governs the convergence of the algorithm. The annealing schedule includes the initial temperature setting, the decrement of the temperature, and the final value of the temperature; at any given temperature, the annealing process proceeds according to the Metropolis algorithm.

The Metropolis algorithm, based on Monte Carlo techniques, developed in 1953, was originally designed to compute the properties of systems in thermal equilibrium. Following is a summary of the algorithm; details can be found, for example, in the original paper [8] by Metropolis et al.

Initially, the system is in state S_0 with an energy $E(S_0)$. In each step of the algorithm, a state S_i of the system is altered to $\tilde{S}_i$ according to the random update scheme, and a resulting change ΔE in the energy of the system is computed:

$$\Delta E = E(\tilde{S}_i) - E(S_i). \tag{4}$$

If $\Delta E \leq 0$, the update is accepted, and the system evolves to the new state $\tilde{S}_i$; if $\Delta E > 0$, the update is accepted with a probability $P(\Delta E)$, where

$$P(\Delta E) = \exp(-\Delta E/T)\,, \tag{5}$$

ant T is the absolute temperature.

The choice of $P(\Delta E)$ ensures that at a temperature T approaching zero, only states with minimum energy have a nonzero probability of occurrence. When the temperature is lowered in a sufficiently slow manner, the system can achieve thermal equilibrium at each temperature, and therefore achieve a minimum energy state at the low final temperature.

2.3 Newton's Method

Newton's method is an iterative method for solving equation systems of the form

$$F(x) = \begin{bmatrix} f_1(x_1, x_2, \ldots, x_n) \\ \vdots \\ f_n(x_1, x_2, \ldots, x_n) \end{bmatrix} = 0. \tag{6}$$

Definition 2.1 *The Jacobian DF of function F in equation (6) is defined by:*

$$DF(x) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \tag{7}$$

Theorem 2.1 *(Newton's Method) Let $F : R^n \longrightarrow R^n$ be continuously differentiable in the neighborhood of ξ where*

$$F(\xi) = 0 \tag{8}$$

Suppose that Jacobian $DF(x)$ is nonsingular at point x. Given a starting point $x_0 \in R^n$, define sequence $x_1, x_2, \ldots,$ of R^n as the following:

$$x_{k+1} = x_k - DF(x_k)^{-1}F(x_k). \tag{9}$$

Then there exists ϵ, $\epsilon > 0$ and for all x, $||x - x_0|| < \epsilon$, there exists $\delta > 0$ such that

$$||x_{k+1} - x|| < \delta ||x_k - x||^2 \tag{10}$$

In other words, sequence (9) converges to ξ quadratically if the initial point x_0 is sufficiently close to ξ.

3 Analytical Apparatus

In this section, we develop analytical tools used in Section 4 in the numerical construction of the quadrature rules. For simplicity, we assume that the integration region Q belongs to R^2; generalization to higher dimensions should be straightforward.

3.1 Notations

Definition 3.1 A *monomial* of order n in R^2 is any term of x and y of the form

$$x^{n_1} y^{n_2} \tag{11}$$

where n_1, n_2 are integers, $0 \le n_1, n_2 \le n$ and $n_1 + n_2 = n$. We denote the set of all monomials of orders less than n by $\mathcal{M}(n)$.

Definition 3.2 The *order* of a quadrature rule is the lowest order of monomials for which the rule is inexact. We denote it by $\mathcal{O}$.

Definition 3.3 The *efficiency* E of a quadrature rule is the ratio of the number of independent monomials (up to a certain order) for which the quadrature rule is exact, to the number of free parameters of the quadrature rule.

Example Suppose that $\mathcal{O}$ is the order of a quadrature rule on an integration region Q, and N is the number of quadrature nodes. Then the number of independent monomials in $\mathcal{M}(\mathcal{O})$ is given by $\frac{\mathcal{O}(\mathcal{O}+1)}{2}$, and the number of ostensibly free parameters for N-point quadrature is $3N$. Therefore, the efficiency E of the quadrature is $\frac{\mathcal{O}(\mathcal{O}+1)/2}{3N}$. For some regions of integration such as the surface of the sphere, the number of natural variables of the polynomials (x, y, z for the spherical shell) is greater than the dimensionality of the region of integration, therefore these relations may be different. In particular, for the spherical shell, $|\mathcal{M}(\mathcal{O})| = \frac{\mathcal{O}(2\mathcal{O}+1)}{2}$ and $E = \frac{\mathcal{O}(2\mathcal{O}+1)}{6N}$.

Definition 3.4 A quadrature on integration region Q is said to be *group invariant* if it is invariant under the transformation of every group element g in Q's symmetry group G.

3.2 Reduction of Dimensionality

Given an integration region Q, we seek quadrature rules of order $\mathcal{O}$ with minimum number of nodes N that possess the following properties:

1. The quadrature rule is group invariant;

2. All weights are positive: quadrature rules with negative weights are unstable with noisy integrands;
3. All quadrature nodes are within the integration region Q (including the boundary).

We evaluate the resulting quadratures according to the efficiency E defined in the preceding section.

Based on the results of Section 2.1, quadrature rules of the form

$$\sum_i w_i \cdot \frac{1}{m} \sum_{l=1}^{m} \Phi_{g_l}(f)(x_i, y_i)$$

automatically integrate correctly all functions not belonging to the identity representation. Thus if we adjust $\{w_i\}$ and $\{(x_i, y_i)\}$ so that the quadrature integrates correctly all polynomials belonging to the identity representation up to a certain degree, the rule will be correct for all polynomials up to that degree. This reduces considerably the number of nonlinear equations one must solve to obtain a quadrature rule.

4 Construction of Quadratures

We now construct group invariant quadrature rules with the mathematical and numerical apparatus developed in Sections 2 and 3.

Remark Given an integration region Q, the symmetry group G is either finite or infinite. If G is finite, we seek quadrature rules that are invariant to the entire group; otherwise, we select some maximal subgroup for which a group invariant quadrature rule exists, and construct quadratures accordingly. In some cases, the size of the symmetry group may grow with the number of nodes in the quadrature; an excellent example of this is the circle, where the order of the maximal subgroup equals the number of nodes N.

Due to Section 2.1, group invariant quadrature nodes may be partitioned into G-orbits where G is the symmetry group. We parameterize the i-th G-orbit by $x(\{\lambda_i\})$, where $\{\lambda_i\} = \lambda_{i1}, \ldots, \lambda_{iu}$, and u is determined by the degrees of freedom of the orbit (eg., $0 \le u \le 2$ if the integration region Q is in R^2). We denote the number of points contained in the i-th orbit by m_i, and the corresponding weight, w_i.

We compute the quadrature nodes $x(\{\lambda_i\})$ and weights w_i by solving the following non-linear system:

$$\begin{aligned} \sum_{i=1}^{A} w_i m_i f_1 \left[x\left(\{\lambda_i\}\right)\right] - I_1 &= 0, \\ \sum_{i=1}^{A} w_i m_i f_2 \left[x\left(\{\lambda_i\}\right)\right] - I_2 &= 0, \\ &\vdots \\ \sum_{i=1}^{A} w_i m_i f_n \left[x\left(\{\lambda_i\}\right)\right] - I_n &= 0. \end{aligned} \tag{12}$$

where A is the number of distinct orbits occupied by the quadrature nodes,

$$I_j \equiv \iint_Q f_j \, dQ, \quad i = 1, 2, \ldots, A, \tag{13}$$

and $f_1, f_2, \dots, f_n$ are the set of polynomials to be evaluated.

Due to Section 2.1, a group invariant quadrature will automatically be correct for any polynomial that is orthogonal to the subspace of group invariant polynomials. Therefore we only need to evaluate polynomials that transform according to the identity representation of G; an appropriate choice of the set of polynomials to be evaluated would be a group invariant orthogonal basis (up to a certain order) on region Q, which may be obtained via equation (3).

We use Newton's method to solve the non-linear system (12), with the iterative sequence defined by equation (9); this process converges quadratically due to Theorem 2.1. Following are the formulae of partial derivatives of individual functions with respect to weights and parameters (of nodes), which are needed in the Newton's method:

$$\frac{\partial f_j}{\partial w_i} = m_i f_j(x(\{\lambda_i\})), \quad i = 1, 2, \dots, A, \tag{14}$$

$$\frac{\partial f_j}{\partial \lambda_{ik}} = w_i \cdot \frac{\partial f_j(x(\{\lambda_i\}))}{\partial \lambda_{ik}} \quad i = 1, 2, \dots, A; \; k = 1, \dots, \mu_i, \tag{15}$$

where μ_i is the number of parameters of the i-th orbit; in the case of T, $\mu_i = 0$, 1, or 2, respectively for orbits containing 1, 3, or 6 points.

As is well-known, Newton's method is extremely sensitive to the choice of the initial approximation x_0 (see Section 2.3). In practice, the non-linear system (12) is often under-constrained: the number of equations that can be solved with weights that are positive and nodes that are in the region of integration is smaller than the number of unknowns $A + \sum_i^A \mu_i$. Simulated annealing provides a tool under such circumstances for finding the initial approximation x_0; we defined the objective function J via the formula:

$$J \equiv \sum_j \frac{1}{j} \left[I_j - \sum_i w_i m_i f_j(x(\{\lambda_i\})) \right]^2 . \tag{16}$$

Our implementation of the method follows closely the standard procedure set forth in [4], with a randomly selected starting configuration S_0, and randomly chosen small displacements of nodes and weights at each step. The decrement of annealing temperature is defined by

$$T_k = \alpha T_{k-1} \tag{17}$$

where α is a constant smaller than but close to 1. Sometimes the cooling process fails, and we need to adjust the temperature manually. Throughout the process, any weight that is negative after the random displacement is set to zero.

5 Quadratures on the Triangle

We have implemented the numerical scheme described in Section 4 on the triangle T (see Figure 2.1) and obtained rules of orders up to 30. Any other triangle may be mapped onto T via an affine transformation.

5.1 Parameterization of Quadrature Nodes

We parameterize a point on the triangle with three dependent variables u_1, u_2, and u_3, with the constraint

$$u_1 + u_2 + u_3 = 1. \tag{18}$$

The variables u_1, u_2, u_3 are related to the Cartesian variables x, y via the following formulae:

$$\begin{aligned}
x &= \frac{2u_1 - u_2 - u_3}{2} && (19)\\
y &= \frac{\sqrt{3}(u_2 - u_3)}{2} && (20)\\
u_1 &= \frac{1 + 2x}{3} && (21)\\
u_2 &= \frac{1 - x + \sqrt{3}y}{3} && (22)\\
u_3 &= \frac{1 - x - \sqrt{3}y}{3}. && (23)
\end{aligned}$$

5.2 Group Invariant Orthogonal Polynomials

Order	Function
0	1
2	$\sqrt{\frac{5}{3}}\,(4x^2 + 4y^2 - 1)$
3	$\left(\frac{2}{3}\right)^{\frac{5}{2}}(4 - 30x^2 + 35x^3 - 30y^2 - 105xy^2)$
4	$5\sqrt{\frac{7}{243}}\left[1 - 12\,(x^2 + y^2) + 36\,(x^2 + y^2)^2 + 16x\,(-x^2 + 3y^2)\right]$
4	$\frac{10}{\sqrt{1917}}\left[26x - 49\,(x^2 - y^2) - 112x\,(x^2 + y^2) + 168\,(x^2 - y^2)\,(x^2 + y^2)\right]$
5	$\frac{4}{9\sqrt{3}}\left[\begin{array}{c} 8 - 140\,(x^2 + y^2) + 420\,(x^2 + y^2)^2 + 70\,x\,(x^2 - 3\,y^2) \\ -385\,x\,(x^2 + y^2)\,(x^2 - 3\,y^2) \end{array}\right]$
6	$\frac{7\sqrt{5}}{9\sqrt{11863}}\left[\begin{array}{c} 9 - 2960\,x\,(x^2 - 3\,y^2) + 10560\,x\,(x^2 - 3\,y^2)\,(x^2 + y^2) \\ -216\,(x^2 + y^2) + 3300\,(x^2 + y^2)^2 - 11440\,(x^2 + y^2)^3 \end{array}\right]$
6	$\frac{7\sqrt{1144}}{243\sqrt{11863}}\left[\begin{array}{c} 140 + 18443\,x^6 - 158205\,x^4\,y^2 + 197685\,x^2\,y^4 - 5283\,y^6 \\ +6680\,x\,(x^2 - 3\,y^2) - 33450\,x\,(x^2 - 3\,y^2)\,(x^2 + y^2) \\ -3360\,(x^2 + y^2) + 11790\,(x^2 + y^2)^2 \end{array}\right]$
7	$\frac{8\sqrt{2}}{243}\left[\begin{array}{c} 40 - 1260\,(x^2 + y^2) + 9240\,(x^2 + y^2)^2 \\ +15015\,x\,(x^2 - 3\,y^2)\,(x^2 + y^2)^2 - 770\,x\,(x^2 - 3\,y^2) \\ -2002\,(11\,x^6 + 15\,x^4\,y^2 + 45\,x^2\,y^4 + 9\,y^6) \end{array}\right]$

Table 1: Group Invariant Orthonormal Polynomials on the Triangle

One set of orthogonal polynomials on T is given by the direct product of Jacobi polynomials and Legendre polynomials:

$$(1-x)^m R_{nm}(\frac{4x-1}{3}) P_m(\frac{\sqrt{3}y}{1-x}) \tag{24}$$

where

$$R_{nm}(z) = P_{n-m}^{2m+1,0}(z)\,. \tag{25}$$

Using the projection operator specified in Section 2.1, we obtain the group invariant orthogonal basis on T; the normalized basis polynomials of orders less than 8 are shown in Table 1.

5.3 Numerical Results

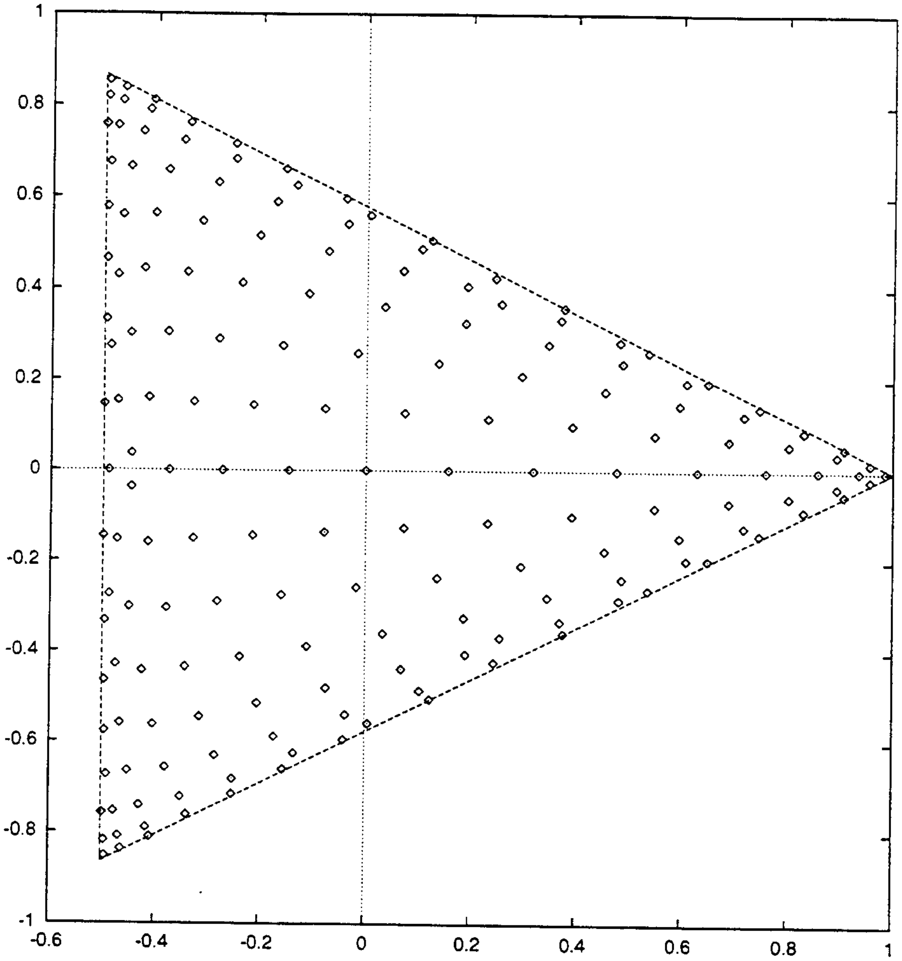


Figure 5.3. Triangle T and the quadrature nodes for $\mathcal{O} = 30$.

In agreement with the criteria specified in Section 3.2, all quadratures we obtained have nodes and weights that are invariant under the action of D_3. In Table 2, we list the quadrature rules of orders $\mathcal{O} = 5, 10, 20$, and 30; the quadrature nodes are partitioned into three types of orbits as described in Section 2.1, and each orbit is listed as a set of parameters $\{\lambda_i\}$, $i = 0, 1, 2$; all weights are normalized so that the integral of any basis polynomial on T is one.

$\mathcal{O}$	Weights (w_i)	Nodes	
		λ_1	λ_2
5	$2.250000000000000E-01$		
	$1.323941527885062E-01$	$5.971587178976982E-02$	
	$1.259391805448271E-01$	$7.974269853530873E-01$	
10	$8.352339980519637E-02$		
	$7.229850592056742E-03$	$4.269134091050350E-03$	
	$7.449217792098051E-02$	$1.439751005418876E-01$	
	$7.864647340310853E-02$	$6.304871745135509E-01$	
	$6.928323087107503E-03$	$9.590375628566449E-01$	
	$6.928323087107503E-03$	$3.500298989727201E-02$	$1.365735762560334E-01$
	$2.951832033477940E-02$	$3.500298989727201E-02$	
	$2.951832033477940E-02$	$3.754907025844263E-02$	$3.327436005886387E-01$
	$3.957936719606124E-02$	$3.754907025844263E-02$	
20	$2.761042699769952E-02$		
	$1.779029547326740E-03$	$1.500649324429017E-03$	
	$2.011239811396117E-02$	$9.413975193895086E-02$	
	$2.681784725933157E-02$	$2.044721240895264E-01$	
	$2.452313380150201E-02$	$4.709995949344253E-01$	
	$1.639457841069538E-02$	$5.779620718158465E-01$	
	$1.479590739864960E-02$	$7.845287856574573E-01$	
	$4.579282277704251E-03$	$9.218618243243946E-01$	
	$1.651826515576217E-03$	$9.776512405413408E-01$	
	$1.651826515576217E-03$	$5.349618187337239E-03$	$6.354966590835223E-02$
	$2.349170908575584E-03$	$5.349618187337239E-03$	
	$2.349170908575584E-03$	$7.954817066198923E-03$	$1.571069189407069E-01$
	$4.465925754181793E-03$	$7.954817066198923E-03$	
	$4.465925754181793E-03$	$1.042239828126384E-02$	$3.956421143643740E-01$
	$6.099566807907971E-03$	$1.042239828126384E-02$	
	$6.099566807907971E-03$	$1.096441479612335E-02$	$2.731675707129105E-01$
	$6.891081327188203E-03$	$1.096441479612335E-02$	
	$6.891081327188203E-03$	$3.856671208546238E-02$	$1.017853824850170E-01$
	$7.997475072478161E-03$	$3.856671208546238E-02$	
	$7.997475072478161E-03$	$3.558050781721823E-02$	$4.466585491764138E-01$
	$7.386134285336023E-03$	$3.558050781721823E-02$	
	$7.386134285336023E-03$	$4.967081636276412E-02$	$1.990107941495031E-01$
	$1.279933187864826E-02$	$4.967081636276412E-02$	
	$1.279933187864826E-02$	$5.851972508433171E-02$	$3.242611836922827E-01$
	$1.725807117569655E-02$	$5.851972508433171E-02$	
	$1.725807117569655E-02$	$1.214977870043943E-01$	$2.085313632101329E-01$
	$1.867294590293547E-02$	$1.214977870043943E-01$	
	$1.867294590293547E-02$	$1.407108449439387E-01$	$3.231705665362575E-01$
	$2.281822405839526E-02$	$1.407108449439387E-01$	
30	$1.557996020289920E-02$		
	$3.177233700534134E-03$	$7.330116432765550E-03$	
	$1.048342663573077E-02$	$8.299567580296455E-02$	
	$1.320945957774363E-02$	$1.509809561254103E-01$	
	$1.497500696627150E-02$	$2.359058598921665E-01$	

$\mathcal{O}$	Weights (w_i)	Nodes	
		λ_1	λ_2
	$1.498790444338419E-02$	$4.380243084078481E-01$	
	$1.333886474102166E-02$	$5.453020482919312E-01$	
	$1.088917111390201E-02$	$6.508817769825403E-01$	
	$8.189440660893461E-03$	$7.534831455971268E-01$	
	$5.575387588607785E-03$	$8.398315422156063E-01$	
	$3.191216473411976E-03$	$9.044510651842024E-01$	
	$1.296715144327045E-03$	$9.565589706397170E-01$	
	$2.982628261349172E-04$	$9.904706447691261E-01$	
	$2.982628261349172E-04$	$9.253711933464866E-04$	$4.152952709133117E-01$
	$9.989056850788964E-04$	$9.253711933464866E-04$	
	$9.989056850788964E-04$	$1.385925855563978E-03$	$6.118990978534904E-02$
	$4.628508491732533E-04$	$1.385925855563978E-03$	
	$4.628508491732533E-04$	$3.682415455910755E-03$	$1.649086901369066E-01$
	$1.234451336382413E-03$	$3.682415455910755E-03$	
	$1.234451336382413E-03$	$3.903223424159366E-03$	$2.503506223200251E-02$
	$5.707198522432062E-04$	$3.903223424159366E-03$	
	$5.707198522432062E-04$	$3.233248155010538E-03$	$3.060644651510958E-01$
	$1.126946125877624E-03$	$3.233248155010538E-03$	
	$1.126946125877624E-03$	$6.467432112236475E-03$	$1.070732837302181E-01$
	$1.747866949407337E-03$	$6.467432112236475E-03$	
	$1.747866949407337E-03$	$3.247475491332623E-03$	$2.299575493455843E-01$
	$1.182818815031656E-03$	$3.247475491332623E-03$	
	$1.182818815031656E-03$	$8.675090806753763E-03$	$3.370366333057830E-01$
	$1.990839294675034E-03$	$8.675090806753763E-03$	
	$1.990839294675034E-03$	$1.559702646731387E-02$	$5.625657618206073E-02$
	$1.900412795035980E-03$	$1.559702646731387E-02$	
	$1.900412795035980E-03$	$1.797672125368521E-02$	$4.024513752124010E-01$
	$4.498365808817451E-03$	$1.797672125368521E-02$	
	$4.498365808817451E-03$	$1.712424535388931E-02$	$2.436547020108285E-01$
	$3.478719460274719E-03$	$1.712424535388931E-02$	
	$3.478719460274719E-03$	$2.288340534658187E-02$	$1.653895856145327E-01$
	$4.102399036723953E-03$	$2.288340534658187E-02$	
	$4.102399036723953E-03$	$3.273759728776665E-02$	$9.930187449584690E-02$
	$4.021761549744162E-03$	$3.273759728776665E-02$	
	$4.021761549744162E-03$	$3.382101234234097E-02$	$3.084783330690550E-01$
	$6.033164660795066E-03$	$3.382101234234097E-02$	
	$6.033164660795066E-03$	$3.554761446001525E-02$	$4.606683185921130E-01$
	$3.946290302129598E-03$	$3.554761446001525E-02$	
	$3.946290302129598E-03$	$5.053979030686655E-02$	$2.188152994539297E-01$
	$6.644044537680268E-03$	$5.053979030686655E-02$	
	$6.644044537680268E-03$	$5.701471491573222E-02$	$3.792095515602741E-01$
	$8.254305856078458E-03$	$5.701471491573222E-02$	
	$8.254305856078458E-03$	$6.415280642120340E-02$	$1.429608194181854E-01$
	$6.496056633406411E-03$	$6.415280642120340E-02$	
	$6.496056633406411E-03$	$8.050114828762564E-02$	$2.837312821059250E-01$
	$9.252778144146602E-03$	$8.050114828762564E-02$	
	$9.252778144146602E-03$	$1.043670681345305E-01$	$1.967374410044408E-01$
	$9.164920726294278E-03$	$1.043670681345305E-01$	
	$9.164920726294278E-03$	$1.138448944287513E-01$	$3.558891412116621E-01$

$\mathcal{O}$	Weights (w_i)	Nodes	
		λ_1	λ_2
	1.156952462809767$E-02$	1.138448944287513$E-01$	
	1.156952462809767$E-02$	1.453634877155238$E-01$	2.598186853519115$E-01$
	1.176111646760917$E-02$	1.453634877155238$E-01$	
	1.176111646760917$E-02$	1.899456528219788$E-01$	3.219231812312984$E-01$
	1.382470218216540$E-02$	1.899456528219788$E-01$	

Table 2: Quadratures on Triangle of Orders $\mathcal{O} = 5, 10, 20$ and 30.

Conversion from the parameters $\{\lambda_i\}$ to $\{u_1, u_2, u_3\}$ is defined by the following rules:

- No λ parameter : $u_1 = u_2 = u_3 = \frac{1}{3}$;
- One parameter λ_1 : $u_1 = \lambda_1, u_2 = u_3 = \frac{1-\lambda_1}{2}$;
- Two parameters λ_1, λ_2: $u_1 = \lambda_1, u_2 = \lambda_2, u_3 = 1 - \lambda_1 - \lambda_2$.

The Cartesian coordinates x, y of each quadrature node in the orbit specified by $\{\lambda_i\}$ may be obtained from any permutation of $\{u_1, u_2, u_3\}$ using formulae (18) and (19).

5.4 Accuracy

Order ($\mathcal{O}$)	Nodes (N)	Error	Order ($\mathcal{O}$)	Nodes (N)	Error
1	1	0	16	54	7.285839$E-16$
2	1	0	17	58	8.721051$E-16$
3	3	1.665335$E-16$	18	66	5.308254$E-16$
4	6	2.081668$E-16$	19	73	8.665267$E-16$
5	7	2.081668$E-16$	20	82	1.081492$E-15$
6	12	2.775558$E-16$	21	85	7.406211$E-16$
7	12	2.914335$E-16$	22	93	7.406211$E-16$
8	15	6.245005$E-16$	23	100	1.256012$E-15$
9	16	2.081668$E-16$	24	106	1.013946$E-15$
10	19	4.293441$E-16$	25	118	1.242504$E-15$
11	25	4.293441$E-16$	26	126	7.236788$E-16$
12	28	4.293441$E-16$	27	138	1.070311$E-15$
13	36	6.314393$E-16$	28	145	1.362410$E-15$
14	40	5.464379$E-16$	29	154	1.057250$E-15$
15	46	6.677055$E-16$	30	184	1.087304$E-15$

Table 3: Errors of Quadrature Rules for Triangles

We test each quadrature of order $\mathcal{O}$ on all monomials in set $\mathcal{M}(\mathcal{O})$; the maximum absolute error for each quadrature is listed in Table 3. These results were obtained with calculations of double precision accuracy.

5.5 Efficiency

Order ($\mathcal{O}$)	Efficiency $E(\%)$	
	Triangle	Tensor Product
2	100.0	100.0
3	66.7	50.0
4	55.6	83.3
5	71.4	55.6
6	58.3	77.8
7	77.8	58.3
8	80.0	75.0
9	93.8	60.0
10	96.5	73.3
11	88.0	61.1
12	92.9	72.1
13	84.0	61.9
14	87.5	71.4
15	87.0	62.5
16	84.0	70.8
17	87.9	63.0
18	86.4	70.3
19	86.8	63.3
20	85.4	70.0
21	90.6	63.6
22	85.4	69.7
23	92.0	63.9
24	94.3	69.4
25	91.8	64.1
26	92.9	69.2
27	91.3	64.3
28	93.3	69.0
29	94.2	64.4
30	84.2	68.9

Table 4: Efficiency of Triangle Rules and Tensor Product Rules

In Table 4, we list the efficiency of each quadrature rule of orders 2 through 30, and that of the corresponding tensor product rules. An analysis of this table reveals that our triangle quadratures tend to be more efficient for higher orders. The efficiency is comparable to the results obtained by Lyness and Jespersen on their rules, whose highest order is twelve; however, their rules tend to be more efficient than ours. The efficiency of our quadratures are better than that of tensor product rules. For a tensor product quadrature rule to be of order $\mathcal{O}$, $\left(\lceil\frac{\mathcal{O}}{2}\rceil\right)^2$ quadrature nodes are needed, yielding an efficiency $E = \frac{\mathcal{O}(\mathcal{O}+1)/2}{3\cdot\lceil\frac{\mathcal{O}}{2}\rceil^2}$ which asymptotically approaches $\frac{2}{3}$.

6 Conclusions

We have presented a numerical scheme combining simple group theory and brute-force optimization to reduce the dimensionality of the nonlinear system used to derive quadrature rules. With this scheme, we obtain quadratures with orders up to 30.

This scheme is readily extensible to other symmetric regions in R^2, and to higher dimensions; one simply has to replace D_3 with the corresponding symmetry groups.

The principal drawback of this scheme is that a significant amount of human intervention is involved in choosing initial points and adjusting the simulated annealing constants. A more systematic procedure would be much desirable. Also, by requiring quadrature rules symmetric to the largest subset of the symmetry group of the integration region, some highly efficient quadratures may be missed by our method. Such cases are observed during our experiments on the triangle; an example is that our scheme will fail to find a 5-point rule that has an order of 4.

7 Acknowledgments

The authors thank Professor Vladimir Rokhlin for his encouragement, advice, and support.

References

[1] J. Berntsen and T. O. Espelid. *Degree 13 Symmetric Quadrature Rules for the Triangle.* Reports in Informatics 44, Dept. of Informatics, University of Bergen, 1990.

[2] M. Dubiner. *Spectral methods on triangles and other domains.* Journal of Scientific Computing, 1991, No. 6, p345-390.

[3] S. Haykin. *Neural Networks — A comprehensive foundation.* 1994. Prentice Hall.

[4] Scott Kirkpatrick, C. D. Gelatt and M. P. Vecchi. 1983. *Optimization by Simulated Annealing.* Science, 1983, V220, p671.

[5] Scott Kirkpatrick. *Optimization by Simulated Annealing: Quantitative Studies.* Journal of Statistical Physics, 1984, V34, p975.

[6] J. N. Lyness and D. Jespersen. *Moderate Degree Symmetric Quadrature Rules for the Triangle.* Journal of the Institute of Mathematics and its Applications, 1975, V15, p19–32.

[7] J. N. Lyness. *On Handling Singularities in Finite Elements.* Numerical Integration — Recent Developments, Software and Applications, p219–233, 1992. Kluwer Academic Publishers, by Terje O. Espelid and Alan Genz.

[8] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta Teller and Edward Teller. *Equation of State Calculations by Fast Computing Machines.* Journal of Chemical Physics, 1953, V21, no. 6, p1087–1092.

[9] A. D. McLaren. *Optimal Numerical Integration on a sphere.* Mathematics of Computation, 1963, Vol. 17, p361–383.

[10] Derek J.S. Robinson. 1995. *A Course in the Theory of Groups.* Springer.

[11] J. Stoer and R. Bulirsche. 1991. *Introduction to Numerical Analysis.* Springer.

[12] Michael Tinkham. 1964. *Group Theory and Quantum Mechanics.* McGraw-Hill Book Company.

[13] E. Wigner. *Group Theory.* Academic Press Inc., New York. 1959.

Quadruple and Octuple Layer Potentials in Two Dimensions I: Analytical Apparatus

P. Kolm and V. Rokhlin
Research Report YALEU/DCS/RR-1176
March 15, 1999

YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

A detailed analysis is presented of all pseudodifferential operators of orders up to 2 encountered in classical potential theory in two dimensions. In a sequel to this paper, the obtained apparatus will be used to construct stable discretizations of arbitrarily high order for a variety of boundary value problems for elliptic partial differential equations.

Quadruple and Octuple Layer Potentials in Two Dimensions I: Analytical Apparatus

P. Kolm and V. Rokhlin

Research Report YALEU/DCS/RR-1176
March 15, 1999

The first author has been supported in part by DARPA/AFOSR under Contract F49620-97-1-0011. The second author has been supported in part by DARPA/AFOSR under Contract F49620-97-1-0011, in part by ONR under grant N00014-96-1-0188, and in part by AFOSR under Contract F49620-97-C-0052.

Approved for public release: distribution is unlimited.

Keywords: *Laplace Equation, Potential Theory, Pseudodifferential Operators, Hypersingular Integral Equations*

1 Introduction

Integral equations of classical potential theory are a tool for the solution of the Laplace equation; they have straightforward analogues for many other elliptic partial differential equations (PDEs). From the point of view of a modern mathematician, they are relatively simple objects. Indeed, a second kind integral equation (SKIE) is a sum of the unity operator and a compact operator; for most practical purposes, such an object behaves like a finite-dimensional system of linear algebraic equations, with the Fredholm alternative replacing the theory of determinants. Integral equations of the first kind (FKIEs) are a considerably more complicated object than those of the second kind. Since a first kind integral operator is compact, solving a first kind integral equation involves the application of the inverse of a compact operator to the right-hand side; depending on the right-hand side, the result might or might not be a function. Since the classical boundary value problems (Dirichlet, Neumann, and Robin) are easily reduced to SKIEs, the original creators of the potential theory simply ignored the FKIEs. Later, FKIEs of classical potential theory have also been investigated, and are now a fairly well-understood object.

In a nutshell, when the solution of a Dirichlet problem is represented by the potential of a single layer, the result is an FKIE; when the solution of a Dirichlet problem is represented by the potential of a double layer, the result is an SKIE. When the solution of a Neumann problem is represented by a single layer potential, the result is an SKIE; and when the solution of a Neumann problem is represented by a double layer potential, the result is not a classical integral equation, but rather an integro-pseudodifferential one (in computational electromagnetics, this particular object is known as a hypersingular equation). Once the integral equation is constructed, the question arises whether it has a solution, whether that solution is unique, etc. Generally, questions of this type are easily answered for the Laplace and Yukawa equations, and less so in other cases.

As a computational tool, SKIEs were popular before the advent of computers; between 1950 and 1970, they were almost completely replaced with Finite Differences and Finite Elements. The only areas where integral equations survived as a numerical tool were those where discretizing the whole area of definition of a PDE is impractical or very difficult, such as the radar scattering and certain areas of aerodynamics. The reasons for this lack of favor have to do with the fact that discretization of most integral equations of potential theory leads to dense systems of linear algebraic equations, while the Finite Elements and Finite Differences result in sparse matrices (hence the name "Finite Elements"). During the last 15 years or so, it has been discovered that many integral operators of potential theory can be applied to arbitrary vectors in a "fast" manner (for a cost proportional to n for the Laplace and Yukawa equations, and for a cost proportional to $n \cdot \log(n)$ for the Helmholtz equation, with n the number of nodes in the discretization of the integral operator). Detailed discussion of such numerical issues is outside the scope of this paper, and we refer the reader to [5, 6]. Here, we remark that the interest in integral formulations of problems of mathematical physics has been increasing, and that classical tools of potential theory turned out to be insufficient for dealing with many problems encountered in practice.

Specifically, many applications lead to integral formulations involving not only integral equations, but also integro-pseudodifferential ones. More frequently, while it is possible to formulate a problem as an FKIE or an SKIE, the numerical behavior (stability) of the resulting schemes leaves much to be desired. In such cases, it is sometimes possible to reformulate the problem as an integro-pseudodifferential equation with drastically improved stability properties (perhaps, after an appropriate preconditioning). A simple example of such a situation is the exterior Neumann problem for the Helmholtz equation, where the classical SKIE has so-called spurious resonances, coinciding with those for the *interior* Dirichlet problem on the same surface, and having nothing to do with the behavior of the exterior Neumann problem being solved. The so-called "combined field equation" solves the problem of spurious resonances at the expense of replacing an integral equation with an integro-pseudodifferential one (see, for example, [1, 12, 14, 17, 20]). Other examples of such situations include problems in scattering theory, in computational elasticity, in fluid dynamics, and in other fields.

In this paper, we investigate in detail the analytical structure of the integro-pseudodifferential equations obtained when Neumann problems are solved via double layer potentials, when Dirichlet problems are solved via quadruple layer potentials, when Neumann problems are solved via quadruple layer potentials, and in several other cases (see (11) - (29) in Section 2 for a detailed list). It turns out that the analytical structure of the obtained equations is quite simple, and involves several standard pseudodifferential operators (derivative, Hilbert transform, derivative of Hilbert transform, inverse of the derivative of the Hilbert transform, and the second derivative), composed (from the left or the right) with simple diagonal operators. We also show that the product of the standard hypersingular integral operator with the standard first kind integral operator of classical potential theory is a second kind integral operator; in other words, these two operators are perfect preconditioners for each other, asymptotically speaking.

Thus, the purpose of this paper is detailed analytical investigation of integro-pseudodifferential operators converting the densities of charge, dipole, quadrupole, and octapole distributions on a smooth curve in $\mathbb{R}^2$ into the potential, normal derivative of the potential, second normal derivative of the potential, and third normal derivative of the potential on that curve. It turns out that each of these operators is a sum of a standard operator (obtained by replacing the curve with a circle), an integral operator with a smooth kernel, and a diagonal operator. Once such expressions are obtained, it is quite easy to construct discretizations of the underlying integro-pseudodifferential operators that are adaptive, stable and of arbitrarily high order. Such discretizations (and resulting PDE solvers) have been constructed and will be reported in a sequel [10] to this paper.

Remark 1.1 *While the results reported here are easily generalized to three dimensions, it should be pointed out that there exist important classes of problems in three dimensions leading to integro-differential equations that are outside the scope of this paper. Specifically, when frequency-domain equations of electromagnetic scattering are reduced to integral equations on the boundary of the scatterer (yielding the so-called Stratton-Chew equations), the resulting integro-pseudodifferential operators are of a type not investigated here (in addition to normal derivatives on the boundary, they involve tangential derivatives); similarly, integral equations* *of elastic (as opposed to acoustic) scattering lead to integral expressions whose analysis is not a straightforward extension of that presented in this paper. Needless to say, such operators are frequently encountered in applications; they are currently under investigation.*

The structure of this paper is as follows. In Section 2, we list the identities that are the purpose of this paper; the remainder of the paper is devoted to proving these identities. In Section 3 the necessary mathematical preliminaries are introduced. In Section 4 we present proofs of some of the results formulated in Section 2; when the proofs of several results are almost identical, we only prove one of them. Finally, in Section 5 we briefly discuss extensions of results of this paper to three dimensions, and to boundary conditions other than Dirichlet, Neumann, and Robin.

Remark 1.2 *The principal purpose of this paper is to present the explicit formulae (50) - (68), (89) - (93), (94) - (99), (100) - (107), to be used in the design of numerical tools for the solution of partial differential equations. The proofs of these formulae in Section 4 below are a fairly standard exercise in classical analysis, provided here for the sake of completeness. The authors expect that many readers will find it unnecessary to read this paper beyond Section 2.*

2 Statement of Results

2.1 Notation

We will be considering Dirichlet and Neumann problems for Laplace's equation in the interior or the exterior of an open region Ω bounded by a Jordan curve $\gamma(t) = (x_1(t), x_2(t))$ in $\mathbb{R}^2$ where $t \in [0, L]$. We will assume that γ is sufficiently smooth, and parametrized by its arclength. The image of γ will be denoted by Γ, so that $\partial\overline{\Omega} = \Gamma$. For a vector $y = (y_1, y_2) \in \mathbb{R}^2$ we will denote its Euclidean norm by $\|y\|$. Further, $c(t)$ will denote the curvature, and $N_\gamma(t)$ or simply $N(t)$, the exterior unit normal to Γ at $\gamma(t)$. Clearly,

$$N(t) = (x_2'(t), -x_1'(t))\,; \tag{1}$$

the situation is illustrated in Fig. 1.

A charge of unit intensity located at the point $x_0 \in \mathbb{R}^2$ generates a potential, $\Phi_{x_0} : \mathbb{R} \setminus \{x_0\} \to \mathbb{R}$, given by the expression

$$\Phi_{x_0}(x) = -\log(\|x - x_0\|)\,, \tag{2}$$

for all $x \neq x_0$. Further, the potential of a unit strength dipole located at $x_0 \in \mathbb{R}^2$, and oriented in the direction $h \in \mathbb{R}^2$, $\|h\| = 1$, is described by the formula

$$\Phi_{x_0,h}(x) = \frac{\langle h, x - x_0\rangle}{\|x - x_0\|^2}\,. \tag{3}$$

As is well known, the potential due to a point charge at $x_0 \in \mathbb{R}^2$, defined by formula (2), is harmonic in any region excluding the source point x_0.

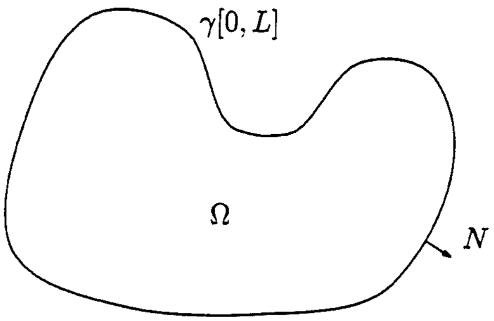


Figure 1: Boundary value problem in $\mathbb{R}^2$.

Definition 2.1 *Suppose that $\sigma : [0,L] \to \mathbb{R}$ is an integrable function. Then we will refer to the functions $p^0_{\gamma,\sigma} : \mathbb{R}^2 \to \mathbb{R}$ and $p^1_{\gamma,\sigma}, p^2_{\gamma,\sigma}, p^3_{\gamma,\sigma} : \mathbb{R}^2 \setminus \Gamma \to \mathbb{R}$, given by the formulae*

$$p^0_{\gamma,\sigma}(x) = \int_0^L \Phi_{\gamma(t)}(x)\,\sigma(t)\,dt\,, \tag{4}$$

$$p^1_{\gamma,\sigma}(x) = \int_0^L \frac{\partial\Phi_{\gamma(t)}(x)}{\partial N(t)}\,\sigma(t)\,dt\,, \tag{5}$$

$$p^2_{\gamma,\sigma}(x) = \int_0^L \frac{\partial^2\Phi_{\gamma(t)}(x)}{\partial N(t)^2}\,\sigma(t)\,dt\,, \tag{6}$$

$$p^3_{\gamma,\sigma}(x) = \int_0^L \frac{\partial^3\Phi_{\gamma(t)}(x)}{\partial N(t)^3}\,\sigma(t)\,dt\,, \tag{7}$$

as the single, double, quadruple and octuple layer potentials, respectively.

Remark 2.1 *The functions $\frac{\partial\Phi_{\gamma(t)}}{\partial N(t)}, \frac{\partial^2\Phi_{\gamma(t)}}{\partial N(t)^2}, \frac{\partial^3\Phi_{\gamma(t)}}{\partial N(t)^3} : \mathbb{R}^2 \setminus \{\gamma(t)\} \to \mathbb{R}$ are often referred to as the dipole-, quadrupole- and octapole potentials, respectively. Obviously,*

$$\frac{\partial\Phi_{\gamma(t)}(x)}{\partial N(t)} = \frac{\langle N(t), x-\gamma(t)\rangle}{\|x-\gamma(t)\|^2}\,, \tag{8}$$

$$\frac{\partial^2\Phi_{\gamma(t)}(x)}{\partial N(t)^2} = \frac{2\langle N(t), x-\gamma(t)\rangle^2}{\|x-\gamma(t)\|^4} - \frac{1}{\|x-\gamma(t)\|^2}\,, \tag{9}$$

$$\frac{\partial^3\Phi_{\gamma(t)}(x)}{\partial N(t)^3} = \frac{8\langle N(t), x-\gamma(t)\rangle^3}{\|x-\gamma(t)\|^6} - \frac{6\,\langle N(t), x-\gamma(t)\rangle}{\|x-\gamma(t)\|^4}\,. \tag{10}$$

Clearly, the potentials $p^1_{\gamma,\sigma}$, $p^2_{\gamma,\sigma}$, $p^3_{\gamma,\sigma}$ are analytic in the interior of Ω for any integrable σ. However, for sufficiently smooth σ and γ, they can be extended to $\overline{\Omega}$ as smooth functions. Similarly, the potentials $p^1_{\gamma,\sigma}$, $p^2_{\gamma,\sigma}$, $p^3_{\gamma,\sigma}$ are analytic functions in the exterior $\mathbb{R}^2 \setminus \overline{\Omega}$ of Ω, and can be extended as smooth functions to $\mathbb{R}^2 \setminus \Omega$. Furthermore, the normal derivatives of these potentials also can be extended up to the boundary as smooth functions. Needless to say, the interior and exterior extensions do not necessarily agree on the boundary Γ (with the obvious exception of $p^{0}_{\gamma,\sigma}(x)$), and we introduce the functions $p^{0,0}_{\gamma,\sigma}$, $p^{1,0}_{\gamma,\sigma,\mathrm{i}}$, $p^{1,0}_{\gamma,\sigma,\mathrm{e}}$, $p^{2,0}_{\gamma,\sigma,\mathrm{i}}$, $p^{2,0}_{\gamma,\sigma,\mathrm{e}}$, $p^{3,0}_{\gamma,\sigma,\mathrm{i}}$, $p^{3,0}_{\gamma,\sigma,\mathrm{e}}$, $p^{0,1}_{\gamma,\sigma,\mathrm{i}}$, $p^{0,1}_{\gamma,\sigma,\mathrm{e}}$, $p^{1,1}_{\gamma,\sigma,\mathrm{i}}$, $p^{1,1}_{\gamma,\sigma,\mathrm{e}}$, $p^{2,1}_{\gamma,\sigma,\mathrm{i}}$, $p^{2,1}_{\gamma,\sigma,\mathrm{e}}$, $p^{0,2}_{\gamma,\sigma,\mathrm{i}}$, $p^{0,2}_{\gamma,\sigma,\mathrm{e}}$, $p^{1,2}_{\gamma,\sigma,\mathrm{i}}$, $p^{1,2}_{\gamma,\sigma,\mathrm{e}}$, $p^{0,3}_{\gamma,\sigma,\mathrm{i}}$, $p^{0,3}_{\gamma,\sigma,\mathrm{e}} : [0,L] \to \mathbb{R}$ via the formulae

$$p^{0,0}_{\gamma,\sigma}(s) = \int_0^L \Phi_{\gamma(t)}(\gamma(s))\,\sigma(t)\,dt\,, \tag{11}$$

$$p^{1,0}_{\gamma,\sigma,\mathrm{i}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial\Phi_{\gamma(t)}(\gamma(s)-h\cdot N(s))}{\partial N(t)}\,\sigma(t)\,dt\,, \tag{12}$$

$$p^{1,0}_{\gamma,\sigma,\mathrm{e}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial\Phi_{\gamma(t)}(\gamma(s)+h\cdot N(s))}{\partial N(t)}\,\sigma(t)\,dt\,, \tag{13}$$

$$p^{2,0}_{\gamma,\sigma,\mathrm{i}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial^2\Phi_{\gamma(t)}(\gamma(s)-h\cdot N(s))}{\partial N(t)^2}\,\sigma(t)\,dt\,, \tag{14}$$

$$p^{2,0}_{\gamma,\sigma,\mathrm{e}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial^2\Phi_{\gamma(t)}(\gamma(s)+h\cdot N(s))}{\partial N(t)^2}\,\sigma(t)\,dt\,, \tag{15}$$

$$p^{3,0}_{\gamma,\sigma,\mathrm{i}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial^3\Phi_{\gamma(t)}(\gamma(s)-h\cdot N(s))}{\partial N(t)^3}\,\sigma(t)\,dt\,, \tag{16}$$

$$p^{3,0}_{\gamma,\sigma,\mathrm{e}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial^3\Phi_{\gamma(t)}(\gamma(s)+h\cdot N(s))}{\partial N(t)^3}\,\sigma(t)\,dt\,, \tag{17}$$

$$p^{0,1}_{\gamma,\sigma,\mathrm{i}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial\Phi_{\gamma(t)}(\gamma(s)-h\cdot N(s))}{\partial N(s)}\,\sigma(t)\,dt\,, \tag{18}$$

$$p^{0,1}_{\gamma,\sigma,\mathrm{e}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial\Phi_{\gamma(t)}(\gamma(s)+h\cdot N(s))}{\partial N(s)}\,\sigma(t)\,dt\,, \tag{19}$$

$$p^{1,1}_{\gamma,\sigma,\mathrm{i}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial^2\Phi_{\gamma(t)}(\gamma(s)-h\cdot N(s))}{\partial N(s)\,\partial N(t)}\,\sigma(t)\,dt\,, \tag{20}$$

$$p^{1,1}_{\gamma,\sigma,\mathrm{e}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial^2\Phi_{\gamma(t)}(\gamma(s)+h\cdot N(s))}{\partial N(s)\,\partial N(t)}\,\sigma(t)\,dt\,, \tag{21}$$

$$p^{2,1}_{\gamma,\sigma,\mathrm{i}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial^3\Phi_{\gamma(t)}(\gamma(s)-h\cdot N(s))}{\partial N(s)\,\partial N(t)^2}\,\sigma(t)\,dt\,, \tag{22}$$

$$p^{2,1}_{\gamma,\sigma,\mathrm{e}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial^3\Phi_{\gamma(t)}(\gamma(s)+h\cdot N(s))}{\partial N(s)\,\partial N(t)^2}\,\sigma(t)\,dt\,, \tag{23}$$

$$p^{0,2}_{\gamma,\sigma,\mathrm{i}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial^2\Phi_{\gamma(t)}(\gamma(s)-h\cdot N(s))}{\partial N(s)^2}\,\sigma(t)\,dt\,, \tag{24}$$

$$p^{0,2}_{\gamma,\sigma,\mathrm{e}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial^2\Phi_{\gamma(t)}(\gamma(s)+h\cdot N(s))}{\partial N(s)^2}\,\sigma(t)\,dt\,, \tag{25}$$

$$p^{1,2}_{\gamma,\sigma,\mathrm{i}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial^3\Phi_{\gamma(t)}(\gamma(s)-h\cdot N(s))}{\partial N(s)^2\,\partial N(t)}\,\sigma(t)\,dt\,, \tag{26}$$

$$p^{1,2}_{\gamma,\sigma,\mathrm{e}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial^2\Phi_{\gamma(t)}(\gamma(s)+h\cdot N(s))}{\partial N(s)^2\,\partial N(t)}\,\sigma(t)\,dt\,, \tag{27}$$

$$p^{0,3}_{\gamma,\sigma,\mathrm{i}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial^3\Phi_{\gamma(t)}(\gamma(s)-h\cdot N(s))}{\partial N(s)^3}\,\sigma(t)\,dt\,, \tag{28}$$

$$p^{0,3}_{\gamma,\sigma,\mathrm{e}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial^3\Phi_{\gamma(t)}(\gamma(s)+h\cdot N(s))}{\partial N(s)^3}\,\sigma(t)\,dt\,. \tag{29}$$

Remark 2.2 *Throughout the paper, the subscripts “i” and “e” will denote the limits from the interior and the exterior towards the boundary, respectively. Furthermore, the superscripts “i, j” (as, for example, in $p^{i,j}_{\gamma,\sigma,\mathrm{e}}(s)$) refers to i times and j times differentiation with respect to $N(t)$ and $N(s)$, respectively.*

Definition 2.2 *Suppose that the function $\sigma : [0,L] \to \mathbb{R}$ is twice continuously differentiable, and that γ is sufficiently smooth. Then we define the operators K^0_γ, $K^{1,0}_{\gamma,\mathrm{i}}$, $K^{1,0}_{\gamma,\mathrm{e}}$, $K^{2,0}_{\gamma,\mathrm{i}}$, $K^{2,0}_{\gamma,\mathrm{e}}$, $K^{3,0}_{\gamma,\mathrm{i}}$, $K^{3,0}_{\gamma,\mathrm{e}}$, $K^{0,1}_{\gamma,\mathrm{i}}$, $K^{0,1}_{\gamma,\mathrm{e}}$, $K^{1,1}_{\gamma,\mathrm{i}}$, $K^{1,1}_{\gamma,\mathrm{e}}$, $K^{2,1}_{\gamma,\mathrm{i}}$, $K^{2,1}_{\gamma,\mathrm{e}}$, $K^{0,2}_{\gamma,\mathrm{i}}$, $K^{0,2}_{\gamma,\mathrm{e}}$, $K^{1,2}_{\gamma,\mathrm{i}}$, $K^{1,2}_{\gamma,\mathrm{e}}$, $K^{0,3}_{\gamma,\mathrm{i}}$, $K^{0,3}_{\gamma,\mathrm{e}} : c^2[0,L] \to c[0,L]$ via the formulae*

$$K^0_\gamma(\sigma)(s) = p^{0,0}_{\gamma,\sigma}(s) = \int_0^L \Phi_{\gamma(t)}(\gamma(s))\,\sigma(t)\,dt\,, \tag{30}$$

$$K^{1,0}_{\gamma,\mathrm{i}}(\sigma)(s) = p^{1,0}_{\gamma,\sigma,\mathrm{i}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial\Phi_{\gamma(t)}(\gamma(s)-h\cdot N(s))}{\partial N(t)}\,\sigma(t)\,dt\,, \tag{31}$$

$$K^{1,0}_{\gamma,\mathrm{e}}(\sigma)(s) = p^{1,0}_{\gamma,\sigma,\mathrm{e}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial\Phi_{\gamma(t)}(\gamma(s)+h\cdot N(s))}{\partial N(t)}\,\sigma(t)\,dt\,, \tag{32}$$

$$K^{2,0}_{\gamma,\mathrm{i}}(\sigma)(s) = p^{2,0}_{\gamma,\sigma,\mathrm{i}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial^2\Phi_{\gamma(t)}(\gamma(s)-h\cdot N(s))}{\partial N(t)^2}\,\sigma(t)\,dt\,, \tag{33}$$

$$K^{2,0}_{\gamma,\mathrm{e}}(\sigma)(s) = p^{2,0}_{\gamma,\sigma,\mathrm{e}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial^2\Phi_{\gamma(t)}(\gamma(s)+h\cdot N(s))}{\partial N(t)^2}\,\sigma(t)\,dt\,, \tag{34}$$

$$K^{3,0}_{\gamma,\mathrm{i}}(\sigma)(s) = p^{3,0}_{\gamma,\sigma,\mathrm{i}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial^3\Phi_{\gamma(t)}(\gamma(s)-h\cdot N(s))}{\partial N(t)^3}\,\sigma(t)\,dt\,, \tag{35}$$

$$K^{3,0}_{\gamma,\mathrm{e}}(\sigma)(s) = p^{3,0}_{\gamma,\sigma,\mathrm{e}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial^3\Phi_{\gamma(t)}(\gamma(s)+h\cdot N(s))}{\partial N(t)^3}\,\sigma(t)\,dt\,, \tag{36}$$

$$K^{0,1}_{\gamma,\mathrm{i}}(\sigma)(s) = p^{0,1}_{\gamma,\sigma,\mathrm{i}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial\Phi_{\gamma(t)}(\gamma(s)-h\cdot N(s))}{\partial N(s)}\,\sigma(t)\,dt\,, \tag{37}$$

$$K^{0,1}_{\gamma,\mathrm{e}}(\sigma)(s) = p^{0,1}_{\gamma,\sigma,\mathrm{e}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial\Phi_{\gamma(t)}(\gamma(s)+h\cdot N(s))}{\partial N(s)}\,\sigma(t)\,dt\,, \tag{38}$$

$$K^{1,1}_{\gamma,\mathrm{i}}(\sigma)(s) = p^{1,1}_{\gamma,\sigma,\mathrm{i}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial^3\Phi_{\gamma(t)}(\gamma(s)-h\cdot N(s))}{\partial N(s)\,\partial N(t)}\,\sigma(t)\,dt\,, \tag{39}$$

$$K^{1,1}_{\gamma,\mathrm{e}}(\sigma)(s) \;=\; p^{1,1}_{\gamma,\sigma,\mathrm{e}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial^3 \Phi_{\gamma(t)}(\gamma(s)+h\cdot N(s))}{\partial N(s)\,\partial N(t)}\,\sigma(t)\,dt\,, \tag{40}$$

$$K^{2,1}_{\gamma,\mathrm{i}}(\sigma)(s) \;=\; p^{2,1}_{\gamma,\sigma,\mathrm{i}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial^3 \Phi_{\gamma(t)}(\gamma(s)-h\cdot N(s))}{\partial N(s)\,\partial N(t)^2}\,\sigma(t)\,dt\,, \tag{41}$$

$$K^{2,1}_{\gamma,\mathrm{e}}(\sigma)(s) \;=\; p^{2,1}_{\gamma,\sigma,\mathrm{e}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial^3 \Phi_{\gamma(t)}(\gamma(s)+h\cdot N(s))}{\partial N(s)\,\partial N(t)^2}\,\sigma(t)\,dt\,, \tag{42}$$

$$K^{0,2}_{\gamma,\mathrm{i}}(\sigma)(s) \;=\; p^{0,2}_{\gamma,\sigma,\mathrm{i}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s)-h\cdot N(s))}{\partial N(s)^2}\,\sigma(t)\,dt\,, \tag{43}$$

$$K^{0,2}_{\gamma,\mathrm{e}}(\sigma)(s) \;=\; p^{0,2}_{\gamma,\sigma,\mathrm{e}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s)+h\cdot N(s))}{\partial N(s)^2}\,\sigma(t)\,dt\,, \tag{44}$$

$$K^{1,2}_{\gamma,\mathrm{i}}(\sigma)(s) \;=\; p^{1,2}_{\gamma,\sigma,\mathrm{i}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial^3 \Phi_{\gamma(t)}(\gamma(s)-h\cdot N(s))}{\partial N(s)^2\,\partial N(t)}\,\sigma(t)\,dt\,, \tag{45}$$

$$K^{1,2}_{\gamma,\mathrm{e}}(\sigma)(s) \;=\; p^{1,2}_{\gamma,\sigma,\mathrm{e}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial^3 \Phi_{\gamma(t)}(\gamma(s)+h\cdot N(s))}{\partial N(s)^2\,\partial N(t)}\,\sigma(t)\,dt\,, \tag{46}$$

$$K^{0,3}_{\gamma,\mathrm{i}}(\sigma)(s) \;=\; p^{0,3}_{\gamma,\sigma,\mathrm{i}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial^3 \Phi_{\gamma(t)}(\gamma(s)-h\cdot N(s))}{\partial N(s)^3}\,\sigma(t)\,dt\,, \tag{47}$$

$$K^{0,3}_{\gamma,\mathrm{e}}(\sigma)(s) \;=\; p^{0,3}_{\gamma,\sigma,\mathrm{e}}(s) = \lim_{h\to 0}\int_0^L \frac{\partial^3 \Phi_{\gamma(t)}(\gamma(s)+h\cdot N(s))}{\partial N(s)^3}\,\sigma(t)\,dt\,. \tag{48}$$

Remark 2.3 *Obviously, the operators* $K^{0,1}_{\gamma,\mathrm{i}}$, $K^{0,1}_{\gamma,\mathrm{e}}$, $K^{0,2}_{\gamma,\mathrm{i}}$, $K^{0,2}_{\gamma,\mathrm{e}}$, $K^{0,3}_{\gamma,\mathrm{i}}$, $K^{0,3}_{\gamma,\mathrm{e}}$, $K^{1,2}_{\gamma,\mathrm{i}}$, $K^{1,2}_{\gamma,\mathrm{e}}$ *given by the formulae (37), (38), (43) - (48) are the adjoints of the operators* $K^{1,0}_{\gamma,\mathrm{i}}$, $K^{1,0}_{\gamma,\mathrm{e}}$, $K^{2,0}_{\gamma,\mathrm{i}}$, $K^{2,0}_{\gamma,\mathrm{e}}$, $K^{3,0}_{\gamma,\mathrm{i}}$, $K^{3,0}_{\gamma,\mathrm{e}}$, $K^{2,1}_{\gamma,\mathrm{i}}$, $K^{2,1}_{\gamma,\mathrm{e}}$ *defined by (31) - (36), (41), (42), respectively. Furthermore,* K^{0}_{γ}, $K^{1,1}_{\gamma,\mathrm{i}}$, $K^{1,1}_{\gamma,\mathrm{e}}$ *defined by (30), (39), (40) are self-adjoint.*

2.2 Physical Interpretation

Formulae (30) – (48) have simple physical interpretations. Specifically, K^0_γ is the linear operator converting a charge distribution on the curve Γ into the potential of that charge distribution on Γ. The operator $K^{1,0}_{\gamma,\mathrm{i}}$ converts a dipole distribution on Γ into the potential created by that distribution on the *inside* of Γ; the operator $K^{1,0}_{\gamma,\mathrm{e}}$ converts a dipole distribution on Γ into the potential created by that distribution on the *outside* of Γ. The operator $K^{0,1}_{\gamma,\mathrm{e}}$ converts a charge distribution on Γ into the normal derivative of the potential created by that distribution on the *outside* of Γ, etc.

Generally, the first superscript denotes the number of differentiations at the source (charges, dipoles, quadrupoles, or octapoles); the second superscript denotes the number of differentiations at the point where the potential is evaluated (potential, normal derivative of the potential, second normal derivative of the potential, third normal derivative of the potential). In agreement with standard practice in the theory of pseudodifferential operators, we will define the order k of either of the operators $K^{i,j}_{\gamma,\mathrm{i}}$ and $K^{i,j}_{\gamma,\mathrm{e}}$ by the formula

$$k = i + j - 1, \tag{49}$$

and observe that in this paper, we describe in detail all operators of potential theory whose order does not exceed 2. For example, we *do* investigate the operator $K^{1,2}_{\gamma,\mathrm{i}}$, converting a dipole distribution on Γ into its second normal derivative, but we *do not* investigate the operator $K^{2,2}_{\gamma,\mathrm{i}}$ converting a quadrupole distribution on Γ into its second normal derivative.

An examination of formulae (50) – (68) shows that the complexity of the expressions describing the operators (30) – (48) on the circle hardly increases as the order of the operator grows. On the other hand, the differences between the operators (30) – (48) on the circle and those on an arbitrary curve become more complicated with the growth of the order of the operator. For example, the operators K^{0}_{γ}, $K^{1,0}_{\gamma,\mathrm{i}}$, $K^{0,1}_{\gamma,\mathrm{i}}$, $K^{1,0}_{\gamma,\mathrm{e}}$, $K^{0,1}_{\gamma,\mathrm{e}}$ on an arbitrary smooth curve always differ from these operators on the circle by a compact operator (see formulae (89) – (93)). Similar differences for the operators $K^{2,0}_{\gamma,\mathrm{i}}$, $K^{2,0}_{\gamma,\mathrm{e}}$, $K^{1,1}_{\gamma,\mathrm{i}}$, $K^{1,1}_{\gamma,\mathrm{e}}$, $K^{0,2}_{\gamma,\mathrm{i}}$, $K^{0,2}_{\gamma,\mathrm{e}}$ involve the curvature of γ (see (94) – (99)). For the operators $K^{3,0}_{\gamma,\mathrm{i}}$, $K^{3,0}_{\gamma,\mathrm{e}}$, $K^{2,1}_{\gamma,\mathrm{i}}$, $K^{2,1}_{\gamma,\mathrm{e}}$, $K^{1,2}_{\gamma,\mathrm{i}}$, $K^{1,2}_{\gamma,\mathrm{e}}$, $K^{0,3}_{\gamma,\mathrm{i}}$, $K^{0,3}_{\gamma,\mathrm{e}}$, the corresponding formulae (100) – (107) already involve the square and the derivative of the curvature, as well as the Hilbert transform of the function.

Remark 2.4 *While it is certainly possible to derive explicit expressions for boundary integral operators of orders higher than* 2, *the complexity of the resulting formulae grows, while their numerical utility decreases. The authors have chosen to draw the line at the order* 2, *mostly because in the applications they anticipate, order* 1 *is sufficient.*

Remark 2.5 *While many of the facts presented in this paper can be obtained "automatically" from the standard theory of pseudodifferential operators, the purpose of this paper is to provide the explicit expressions (50) - (68) to be used in numerical calculations. Thus, we are ignoring the connections between the formulae (50) - (68), (89) - (93), (94) - (99), (100) - (107), and the more general theory of pseudodifferential operators.*

2.3 Results

The limits (12), (13), (18), (19) have been studied in detail in the literature (see, for example, [13, 11]). In Section 4, we conduct a similar investigation of (14) – (17), (20) – (29); first for a circle, and then for a sufficiently smooth Jordan curve. The following theorem provides explicit expressions for the operators (30) – (48) on the circle.

Theorem 2.6 *Suppose that γ is a circle of radius r parametrized by its arclength with the exterior unit normal denoted by N, k is an arbitrary integer, and $s \in [-\pi r, \pi r]$. Then,*

$$\begin{aligned} (a) \qquad K^{0}_{\gamma}(e^{ikt/r})(s) &= p^{0,0}_{\gamma,\, e^{ikt/r}}(s) = \int_{-\pi r}^{\pi r} \Phi_{\gamma(t)}(\gamma(s))\, e^{ikt/r}\, dt \\ &= \begin{cases} \pi\, |k|^{-1}\, r\, e^{iks/r}, & \text{for } k \neq 0, \\ -2\,\pi\, r\, \log(r), & \text{for } k = 0, \end{cases} \end{aligned} \tag{50}$$

(b)

$$\begin{aligned} K^{1,0}_{\gamma,\mathrm{i}}(e^{ikt/r})(s) &= p^{1,0}_{\gamma,\, e^{ikt/r},\mathrm{i}}(s) = \lim_{h\to 0}\int_{-\pi r}^{\pi r} \frac{\partial \Phi_{\gamma(t)}(\gamma(s)-h\cdot N(s))}{\partial N(t)}\, e^{ikt/r}\, dt \\ &= \begin{cases} -\pi\, e^{iks/r}\,, & \text{for } k\neq 0\,, \\ -2\,\pi\,, & \text{for } k=0\,, \end{cases} \end{aligned} \tag{51}$$

$$\begin{aligned} K^{1,0}_{\gamma,\mathrm{e}}(e^{ikt/r})(s) &= p^{1,0}_{\gamma,\, e^{ikt/r},\mathrm{e}}(s) = \lim_{h\to 0}\int_{-\pi r}^{\pi r} \frac{\partial \Phi_{\gamma(t)}(\gamma(s)+h\cdot N(s))}{\partial N(t)}\, e^{ikt/r}\, dt \\ &= \begin{cases} \pi\, e^{iks/r}\,, & \text{for } k\neq 0\,, \\ 0\,, & \text{for } k=0\,, \end{cases} \end{aligned} \tag{52}$$

(c)

$$\begin{aligned} K^{2,0}_{\gamma,\mathrm{i}}(e^{ikt/r})(s) &= p^{2,0}_{\gamma,\, e^{ikt/r},\mathrm{i}}(s) = \lim_{h\to 0}\int_{-\pi r}^{\pi r} \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s)-h\cdot N(s))}{\partial N(t)^2}\, e^{ikt/r}\, dt \\ &= \begin{cases} \pi\,(|k|+1)\, r^{-1}\, e^{iks/r}\,, & \text{for } k\neq 0\,, \\ 2\,\pi\, r^{-1}\,, & \text{for } k=0\,, \end{cases} \end{aligned} \tag{53}$$

$$\begin{aligned} K^{2,0}_{\gamma,\mathrm{e}}(e^{ikt/r})(s) &= p^{2,0}_{\gamma,\, e^{ikt/r},\mathrm{e}}(s) = \lim_{h\to 0}\int_{-\pi r}^{\pi r} \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s)+h\cdot N(s))}{\partial N(t)^2}\, e^{ikt/r}\, dt \\ &= \begin{cases} \pi\,(|k|-1)\, r^{-1}\, e^{iks/r}\,, & \text{for } k\neq 0\,, \\ 0\,, & \text{for } k=0\,, \end{cases} \end{aligned} \tag{54}$$

(d)

$$\begin{aligned} K^{3,0}_{\gamma,\mathrm{i}}(e^{ikt/r})(s) &= p^{3,0}_{\gamma,\, e^{ikt/r},\mathrm{i}}(s) = \lim_{h\to 0}\int_{-\pi r}^{\pi r} \frac{\partial^3 \Phi_{\gamma(t)}(\gamma(s)-h\cdot N(s))}{\partial N(t)^3}\, e^{ikt/r}\, dt \\ &= \begin{cases} -\pi\,(|k|+1)\,(|k|+2)\, r^{-2}\, e^{iks/r}\,, & \text{for } k\neq 0\,, \\ -4\,\pi\, r^{-2}\,, & \text{for } k=0\,, \end{cases} \end{aligned} \tag{55}$$

$$\begin{aligned} K^{3,0}_{\gamma,\mathrm{e}}(e^{ikt/r})(s) &= p^{3,0}_{\gamma,\, e^{ikt/r},\mathrm{e}}(s) = \lim_{h\to 0}\int_{-\pi r}^{\pi r} \frac{\partial^3 \Phi_{\gamma(t)}(\gamma(s)+h\cdot N(s))}{\partial N(t)^3}\, e^{ikt/r}\, dt \\ &= \begin{cases} \pi\,(|k|-1)\,(|k|-2)\, r^{-2}\, e^{iks/r}\,, & \text{for}k\neq 0\,, \\ 0\,, & \text{for } k=0\,, \end{cases} \end{aligned} \tag{56}$$

(e)

$$\begin{aligned} K^{0,1}_{\gamma,\mathrm{i}}(e^{ikt/r})(s) &= p^{0,1}_{\gamma,\, e^{ikt/r},\mathrm{i}}(s) = \lim_{h\to 0}\int_{-\pi r}^{\pi r} \frac{\partial \Phi_{\gamma(t)}(\gamma(s)-h\cdot N(s))}{\partial N(s)}\, e^{ikt/r}\, dt \\ &= \begin{cases} \pi\, e^{iks/r}\,, & \text{for } k\neq 0\,, \\ 0\,, & \text{for } k=0\,, \end{cases} \end{aligned} \tag{57}$$

$$\begin{aligned} K^{0,1}_{\gamma,\mathrm{e}}(e^{ikt/r})(s) &= p^{0,1}_{\gamma,\, e^{ikt/r},\mathrm{e}}(s) = \lim_{h\to 0}\int_{-\pi r}^{\pi r} \frac{\partial \Phi_{\gamma(t)}(\gamma(s)+h\cdot N(s))}{\partial N(s)}\, e^{ikt/r}\, dt \\ &= \begin{cases} -\pi\, e^{iks/r}\,, & \text{for } k\neq 0\,, \\ -2\,\pi\,, & \text{for } k=0\,, \end{cases} \end{aligned} \tag{58}$$

(f)

$$\begin{aligned} K^{1,1}_{\gamma,\mathrm{i}}(e^{ikt/r})(s) &= p^{1,1}_{\gamma,e^{ikt/r},\mathrm{i}}(s) = \lim_{h\to 0}\int_{-\pi r}^{\pi r} \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s) - h\cdot N(s))}{\partial N(s)\,\partial N(t)}\, e^{ikt/r}\, dt \\ &= \begin{cases} -\pi\,|k|\,r^{-1}\,e^{iks/r}\,, & \text{for } k \neq 0\,, \\ 0\,, & \text{for } k = 0\,, \end{cases} \end{aligned} \tag{59}$$

$$\begin{aligned} K^{1,1}_{\gamma,\mathrm{e}}(e^{ikt/r})(s) &= p^{1,1}_{\gamma,e^{ikt/r},\mathrm{e}}(s) = \lim_{h\to 0}\int_{-\pi r}^{\pi r} \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s) + h\cdot N(s))}{\partial N(s)\,\partial N(t)}\, e^{ikt/r}\, ds \\ &= \begin{cases} -\pi\,|k|\,r^{-1}\,e^{iks/r}\,, & \text{for } k \neq 0\,, \\ 0\,, & \text{for } k = 0\,, \end{cases} \end{aligned} \tag{60}$$

(g)

$$\begin{aligned} K^{2,1}_{\gamma,\mathrm{i}}(e^{ikt/r})(s) &= p^{2,1}_{\gamma,e^{ikt/r},\mathrm{i}}(s) = \lim_{h\to 0}\int_{-\pi r}^{\pi r} \frac{\partial^3 \Phi_{\gamma(t)}(\gamma(s) - h\cdot N(s))}{\partial N(s)\,\partial N(t)^2}\, e^{ikt/r}\, dt \\ &= \begin{cases} \pi\,|k|\,(|k|+1)\,r^{-2}\,e^{iks/r}\,, & \text{for } k \neq 0\,, \\ 0\,, & \text{for } k = 0\,, \end{cases} \end{aligned} \tag{61}$$

$$\begin{aligned} K^{2,1}_{\gamma,\mathrm{e}}(e^{ikt/r})(s) &= p^{2,1}_{\gamma,e^{ikt/r},\mathrm{e}}(s) = \lim_{h\to 0}\int_{-\pi r}^{\pi r} \frac{\partial^3 \Phi_{\gamma(t)}(\gamma(s) + h\cdot N(s))}{\partial N(s)\,\partial N(t)^2}\, e^{ikt/r}\, dt \\ &= \begin{cases} -\pi\,|k|\,(|k|-1)\,r^{-2}\,e^{iks/r}\,, & \text{for } k \neq 0\,, \\ 0\,, & \text{for } k = 0\,, \end{cases} \end{aligned} \tag{62}$$

(h)

$$\begin{aligned} K^{0,2}_{\gamma,\mathrm{i}}(e^{ikt/r})(s) &= p^{0,2}_{\gamma,e^{ikt/r},\mathrm{i}}(s) = \lim_{h\to 0}\int_{-\pi r}^{\pi r} \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s) - h\cdot N(s))}{\partial N(s)^2}\, e^{ikt/r}\, dt \\ &= \begin{cases} \pi\,(|k|-1)\,r^{-1}\,e^{iks/r}\,, & \text{for } k \neq 0\,, \\ 0\,, & \text{for } k = 0\,, \end{cases} \end{aligned} \tag{63}$$

$$\begin{aligned} K^{0,2}_{\gamma,\mathrm{e}}(e^{ikt/r})(s) &= p^{0,2}_{\gamma,e^{ikt/r},\mathrm{e}}(s) = \lim_{h\to 0}\int_{-\pi r}^{\pi r} \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s) + h\cdot N(s))}{\partial N(s)^2}\, e^{ikt/r}\, dt \\ &= \begin{cases} \pi\,(|k|+1)\,r^{-1}\,e^{iks/r}\,, & \text{for } k \neq 0\,, \\ 2\,\pi\,r^{-1}\,, & \text{for } k = 0\,, \end{cases} \end{aligned} \tag{64}$$

(i)

$$\begin{aligned} K^{1,2}_{\gamma,\mathrm{i}}(e^{ikt/r})(s) &= p^{1,2}_{\gamma,e^{ikt/r},\mathrm{i}}(s) = \lim_{h\to 0}\int_{-\pi r}^{\pi r} \frac{\partial^3 \Phi_{\gamma(t)}(\gamma(s) - h\cdot N(s))}{\partial N(s)^2\,\partial N(t)}\, e^{ikt/r}\, dt \\ &= \begin{cases} -\pi\,|k|\,(|k|-1)\,r^{-2}\,e^{iks/r}\,, & \text{for } k \neq 0\,, \\ 0\,, & \text{for } k = 0\,, \end{cases} \end{aligned} \tag{65}$$

$$\begin{aligned} K^{1,2}_{\gamma,\mathrm{e}}(e^{ikt/r})(s) &= p^{1,2}_{\gamma,e^{ikt/r},\mathrm{e}}(s) = \lim_{h\to 0}\int_{-\pi r}^{\pi r} \frac{\partial^3 \Phi_{\gamma(t)}(\gamma(s) + h\cdot N(s))}{\partial N(s)^2\,\partial N(t)}\, e^{ikt/r}\, dt \\ &= \begin{cases} \pi\,|k|\,(|k|+1)\,r^{-2}\,e^{iks/r}\,, & \text{for } k \neq 0\,, \\ 0\,, & \text{for } k = 0\,, \end{cases} \end{aligned} \tag{66}$$

$$
\begin{aligned}
\textit{(j)}\qquad K^{0,3}_{\gamma,\mathrm{i}}(e^{ikt/r})(s) &= p^{0,3}_{\gamma,\,e^{ikt/r},\,\mathrm{i}}(s) = \lim_{h\to 0}\int_{-\pi r}^{\pi r} \frac{\partial^3 \Phi_{\gamma(t)}(\gamma(s) - h\cdot N(s))}{\partial N(s)^3}\, e^{ikt/r}\, dt \\
&= \begin{cases} \pi\,(|k|-1)\,(|k|-2)\,r^{-2}\,e^{iks/r}\,, & \text{for } k \neq 0\,, \\ 0\,, & \text{for } k = 0\,, \end{cases}
\end{aligned} \tag{67}
$$

$$
\begin{aligned}
K^{0,3}_{\gamma,\mathrm{e}}(e^{ikt/r})(s) &= p^{0,3}_{\gamma,\,e^{ikt/r},\,\mathrm{e}}(s) = \lim_{h\to 0}\int_{-\pi r}^{\pi r} \frac{\partial^3 \Phi_{\gamma(t)}(\gamma(s) + h\cdot N(s))}{\partial N(s)^3}\, e^{ikt/r}\, dt \\
&= \begin{cases} -\pi\,(|k|+1)\,(|k|+2)\,r^{-2}\,e^{iks/r}\,, & \text{for } k \neq 0\,, \\ -4\,\pi\,r^{-2}\,, & \text{for } k = 0\,. \end{cases}
\end{aligned} \tag{68}
$$

Formulae (50) – (68) describe the action of the operators (30) – (48) on the circle for functions of the form $e^{ikt/r}$, with $k = 0, \pm 1, \pm 2, \ldots$ Now, it immediately follows from (50) – (68) that for any periodic function $\sigma : [0, L] \to \mathbb{C}$ given by its Fourier series

$$
\sigma(t) = \sum_{k=-\infty}^{\infty} \widehat{\sigma}_k\, e^{2\pi i k t/L}\,, \tag{69}
$$

the operators (30) – (48) (γ is the circle of radius $r = \frac{L}{2\pi}$) assume the form

$$
\text{(a)}\qquad K^{0}_{\gamma}(\sigma)(s) = -L\,\log\!\left(\frac{L}{2\,\pi}\right)\widehat{\sigma}_0 + \frac{L}{2}\sum_{\substack{k=-\infty \\ k\neq 0}}^{\infty} \frac{1}{|k|}\widehat{\sigma}_k e^{2\pi i k s/L}\,, \tag{70}
$$

$$
\begin{aligned}
\text{(b)}\qquad K^{1,0}_{\gamma,\mathrm{i}}(\sigma)(s) &= -2\,\pi\,\widehat{\sigma}_0 - \pi \sum_{\substack{k=-\infty \\ k\neq 0}}^{\infty} \widehat{\sigma}_k\, e^{2\pi i k s/L} \\
&= -\pi\,\sigma(s) - \pi\,\widehat{\sigma}_0\,,
\end{aligned} \tag{71}
$$

$$
\begin{aligned}
K^{1,0}_{\gamma,\mathrm{e}}(\sigma)(s) &= \pi \sum_{\substack{k=-\infty \\ k\neq 0}}^{\infty} \widehat{\sigma}_k\, e^{2\pi i k s/L} \\
&= \pi\,\sigma(s) - \pi\,\widehat{\sigma}_0\,,
\end{aligned} \tag{72}
$$

$$
\begin{aligned}
\text{(c)}\qquad K^{2,0}_{\gamma,\mathrm{i}}(\sigma)(s) &= \frac{4\,\pi^2}{L}\,\widehat{\sigma}_0 + \frac{2\,\pi^2}{L}\sum_{\substack{k=-\infty \\ k\neq 0}}^{\infty} (|k|+1)\,\widehat{\sigma}_k\, e^{2\pi i k s/L} \\
&= \frac{2\,\pi^2}{L}\,\sigma(s) + \pi\,H(\sigma')(s) + \frac{2\,\pi^2}{L}\,\widehat{\sigma}_0\,,
\end{aligned} \tag{73}
$$

$$
\begin{aligned}
K^{2,0}_{\gamma,\mathrm{e}}(\sigma)(s) &= \frac{2\,\pi^2}{L}\sum_{\substack{k=-\infty \\ k\neq 0}}^{\infty} (|k|-1)\,\widehat{\sigma}_k\, e^{2\pi i k s/L} \\
&= -\frac{2\,\pi^2}{L}\,\sigma(s) + \pi\,H(\sigma')(s) + \frac{2\,\pi^2}{L}\,\widehat{\sigma}_0\,,
\end{aligned} \tag{74}
$$

(d)

$$\begin{aligned} K^{3,0}_{\gamma,\mathrm{i}}(\sigma)(s) &= -\frac{16\,\pi^3}{L^2}\,\widehat{\sigma}_0 - \frac{4\,\pi^3}{L^2}\sum_{\substack{k=-\infty\\k\neq 0}}^{\infty}(|k|+1)\,(|k|+2)\,\widehat{\sigma}_k\,e^{2\pi i k s/L} \\ &= -\frac{8\,\pi^3}{L^2}\,\sigma(s) + \pi\,\sigma''(s) - \frac{6\,\pi^2}{L}\,H(\sigma')(s) - \frac{8\,\pi^3}{L^2}\,\widehat{\sigma}_0\,, \end{aligned} \tag{75}$$

$$\begin{aligned} K^{3,0}_{\gamma,\mathrm{e}}(\sigma)(s) &= \frac{4\,\pi^3}{L^2}\sum_{\substack{k=-\infty\\k\neq 0}}^{\infty}(|k|-1)\,(|k|-2)\,\widehat{\sigma}_k\,e^{2\pi i k s/L} \\ &= \frac{8\,\pi^3}{L^2}\,\sigma(s) - \pi\,\sigma''(s) - \frac{6\,\pi^2}{L}\,H(\sigma')(s) - \frac{8\,\pi^3}{L^2}\,\widehat{\sigma}_0\,, \end{aligned} \tag{76}$$

(e)

$$\begin{aligned} K^{0,1}_{\gamma,\mathrm{i}}(\sigma)(s) &= \pi\sum_{\substack{k=-\infty\\k\neq 0}}^{\infty}\widehat{\sigma}_k\,e^{2\pi i k s/L} \\ &= \pi\,\sigma(s) - \pi\,\widehat{\sigma}_0\,, \end{aligned} \tag{77}$$

$$\begin{aligned} K^{0,1}_{\gamma,\mathrm{e}}(\sigma)(s) &= -2\,\pi\,\widehat{\sigma}_0 - \pi\sum_{\substack{k=-\infty\\k\neq 0}}^{\infty}\widehat{\sigma}_k\,e^{2\pi i k s/L} \\ &= -\pi\,\sigma(s) - \pi\,\widehat{\sigma}_0\,, \end{aligned} \tag{78}$$

(f)

$$\begin{aligned} K^{1,1}_{\gamma,\mathrm{i}}(\sigma)(s) &= -\frac{2\,\pi^2}{L}\sum_{\substack{k=-\infty\\k\neq 0}}^{\infty}|k|\,\widehat{\sigma}_k\,e^{iks/L} \\ &= -\pi\,H(\sigma')(s)\,, \end{aligned} \tag{79}$$

$$\begin{aligned} K^{1,1}_{\gamma,\mathrm{e}}(\sigma)(s) &= -\frac{2\,\pi^2}{L}\sum_{\substack{k=-\infty\\k\neq 0}}^{\infty}|k|\,\widehat{\sigma}_k\,e^{iks/L} \\ &= -\pi\,H(\sigma')(s)\,, \end{aligned} \tag{80}$$

(g)

$$\begin{aligned} K^{2,1}_{\gamma,\mathrm{i}}(\sigma)(s) &= \frac{4\,\pi^3}{L^2}\sum_{\substack{k=-\infty\\k\neq 0}}^{\infty}|k|\,(|k|+1)\,\widehat{\sigma}_k\,e^{2\pi i k s/L} \\ &= -\pi\,\sigma''(s) + \frac{2\,\pi^2}{L}\,H(\sigma')(s)\,, \end{aligned} \tag{81}$$

$$\begin{aligned} K^{2,1}_{\gamma,\mathrm{e}}(\sigma)(s) &= -\frac{4\,\pi^3}{L^2}\sum_{\substack{k=-\infty\\k\neq 0}}^{\infty}|k|\,(|k|-1)\,\widehat{\sigma}_k\,e^{2\pi i k s/L} \\ &= \pi\,\sigma''(s) + \frac{2\,\pi^2}{L}\,H(\sigma')(s)\,, \end{aligned} \tag{82}$$

(h)

$$K^{0,2}_{\gamma,\mathrm{i}}(\sigma)(s) = \frac{2\,\pi^2}{L}\sum_{\substack{k=-\infty\\k\neq 0}}^{\infty}(|k|-1)\,\widehat{\sigma}_k\,e^{2\pi i k s/L}$$

$$= -\frac{2\,\pi^2}{L}\,\sigma(s) + \pi\,H(\sigma')(s) + \frac{2\,\pi^2}{L}\,\widehat{\sigma}_0\,, \tag{83}$$

$$\begin{aligned} K^{0,2}_{\gamma,\mathrm{e}}(\sigma)(s) &= \frac{4\,\pi^2}{L}\,\widehat{\sigma}_0 + \frac{2\,\pi^2}{L}\sum_{\substack{k=-\infty\\k\neq 0}}^{\infty}(|k|+1)\,\widehat{\sigma}_k\,e^{2\pi iks/L} \\ &= \frac{2\,\pi^2}{L}\,\sigma(s) + \pi\,H(\sigma')(s) + \frac{2\,\pi^2}{L}\,\widehat{\sigma}_0\,, \end{aligned} \tag{84}$$

(i)

$$\begin{aligned} K^{1,2}_{\gamma,\mathrm{i}}(\sigma)(s) &= -\frac{4\,\pi^3}{L^2}\sum_{\substack{k=-\infty\\k\neq 0}}^{\infty}|k|\,(|k|-1)\,\widehat{\sigma}_k\,e^{2\pi iks/L} \\ &= \pi\,\sigma''(s) + \frac{2\,\pi^2}{L}\,H(\sigma')(s)\,, \end{aligned} \tag{85}$$

$$\begin{aligned} K^{1,2}_{\gamma,\mathrm{e}}(\sigma)(s) &= \frac{4\,\pi^3}{L^2}\sum_{\substack{k=-\infty\\k\neq 0}}^{\infty}|k|\,(|k|+1)\,\widehat{\sigma}_k\,e^{2\pi iks/L} \\ &= -\pi\,\sigma''(s) + \frac{2\,\pi^2}{L}\,H(\sigma')(s)\,, \end{aligned} \tag{86}$$

(j)

$$\begin{aligned} K^{0,3}_{\gamma,\mathrm{i}}(\sigma)(s) &= \frac{4\,\pi^3}{L^2}\sum_{\substack{k=-\infty\\k\neq 0}}^{\infty}(|k|-1)\,(|k|-2)\,\widehat{\sigma}_k\,e^{2\pi iks/L} \\ &= \frac{8\,\pi^3}{L^2}\,\sigma(s) - \pi\,\sigma''(s) - \frac{6\,\pi^2}{L}\,H(\sigma')(s) - \frac{8\,\pi^3}{L^2}\,\widehat{\sigma}_0\,, \end{aligned} \tag{87}$$

$$\begin{aligned} K^{0,3}_{\gamma,\mathrm{e}}(\sigma)(s) &= -\frac{16\,\pi^3}{L^2}\,\widehat{\sigma}_0 - \frac{4\,\pi^3}{L^2}\sum_{\substack{k=-\infty\\k\neq 0}}^{\infty}(|k|+1)\,(|k|+2)\,\widehat{\sigma}_k\,e^{2\pi iks/L} \\ &= -\frac{8\,\pi^3}{L^2}\,\sigma(s) + \pi\,\sigma''(s) - \frac{6\,\pi^2}{L}\,H(\sigma')(s) - \frac{8\,\pi^3}{L^2}\,\widehat{\sigma}_0\,, \end{aligned} \tag{88}$$

with $\widehat{\sigma}_k$ denoting the k-th Fourier coefficient of the function σ, and H the Hilbert transform (see (130) in Section 3.3).

Theorem 2.6 above is proved by direct evaluation of the relevant integrals (in Section 4 below, we compute these integrals via the theory of residues). Formulae (70) – (88) are an immediate consequence of Theorem 2.6; they provide explicit expressions for the operators (30) – (48) when γ is a circle.

The following theorem follows easily from well-known results (see, for example, [19, 13]), here stated in a slightly different form.

Theorem 2.7 *Suppose that $\gamma : [0, L] \to \mathbb{R}^2$ is a k times continuously differentiable Jordan curve parametrized by its arclength, and that $\eta : [0, L] \to \mathbb{R}^2$ denotes the circle of radius r. Then, for any sufficiently smooth function $\sigma : [0, L] \to \mathbb{R}$,*

(a)

$$K^0_\gamma(\sigma)(s) = K^0_\eta(\sigma)(s) + M_0(\sigma)(s)\,, \tag{89}$$

$$
\begin{aligned}
(b) \qquad K^{1,0}_{\gamma,\mathrm{i}}(\sigma)(s) &= K^{1,0}_{\eta,\mathrm{i}}(\sigma)(s) + M_1(\sigma)(s) \\
&= -\pi\,\sigma(s) + N_1(\sigma)(s)\,, \qquad (90) \\
K^{1,0}_{\gamma,\mathrm{e}}(\sigma)(s) &= K^{1,0}_{\eta,\mathrm{e}}(\sigma)(s) + M_1(\sigma)(s) \\
&= \pi\,\sigma(s) + N_1(\sigma)(s)\,, \qquad (91) \\
(c) \qquad K^{0,1}_{\gamma,\mathrm{i}}(\sigma)(s) &= K^{0,1}_{\eta,\mathrm{i}}(\sigma)(s) + M_1^{\star}(\sigma)(s) \\
&= \pi\,\sigma(s) + N_1^{\star}(\sigma)(s)\,, \qquad (92) \\
K^{0,1}_{\gamma,\mathrm{e}}(\sigma)(s) &= K^{0,1}_{\eta,\mathrm{e}}(\sigma)(s) + M_1^{\star}(\sigma)(s) \\
&= -\pi\,\sigma(s) + N_1^{\star}(\sigma)(s)\,, \qquad (93)
\end{aligned}
$$

where $M_0, M_1, N_1 : c[0,L] \to c[0,L]$ *are integral operators with kernels* $m_0(s,t) \in c^{k-1}([0,L] \times [0,L])$, $m_1(s,t)$, $n_1(s,t) \in c^{k-2}([0,L] \times [0,L])$, *respectively. Furthermore,* $M_1^{\star}$, $N_1^{\star}$ *are the adjoints of* M_1, N_1, *respectively, and the operator* M_0 *is self-adjoint.*

Theorem 2.7 approximates the operators K^0_{γ}, $K^{1,0}_{\gamma,\mathrm{i}}$, $K^{1,0}_{\gamma,\mathrm{e}}$, $K^{0,1}_{\gamma,\mathrm{i}}$, $K^{0,1}_{\gamma,\mathrm{e}}$ for an arbitrary smooth Jordan curve by the same operators on the circle; Theorem 2.8 below extends these results to the operators (33), (34), (39), (40), (43), (44). While Theorem 2.7 is well-known, the authors failed to find Theorem 2.8 in the literature.

Theorem 2.8 *Suppose that* $\gamma : [0,L] \to \mathbb{R}^2$ *is a k times continuously differentiable Jordan curve parametrized by its arclength, and that* $\eta : [0,L] \to \mathbb{R}^2$ *denotes the circle of radius* $\frac{L}{2\pi}$, *also parametrized by its arclength. Then, for any sufficiently smooth function* $\sigma : [0,L] \to \mathbb{R}$,

$$
\begin{aligned}
(a) \qquad K^{2,0}_{\gamma,\mathrm{i}}(\sigma)(s) &= \left(\pi\, c(s) - \frac{2\,\pi^2}{L}\right)\sigma(s) + K^{2,0}_{\eta,\mathrm{i}}(\sigma)(s) + M_2(\sigma)(s) \\
&= \pi\, c(s)\,\sigma(s) + \pi\, H(\sigma')(s) + N_2(\sigma)(s)\,, \qquad (94) \\
K^{2,0}_{\gamma,\mathrm{e}}(\sigma)(s) &= -\left(\pi\, c(s) - \frac{2\,\pi^2}{L}\right)\sigma(s) + K^{2,0}_{\eta,\mathrm{e}}(\sigma)(s) + M_2(\sigma)(s) \\
&= -\pi\, c(s)\,\sigma(s) + \pi\, H(\sigma')(s) + N_2(\sigma)(s)\,, \qquad (95) \\
(b) \qquad K^{1,1}_{\gamma,\mathrm{i}}(\sigma)(s) &= K^{1,1}_{\eta,\mathrm{i}}(\sigma)(s) + G_2(\sigma)(s) \\
&= -\pi\, H(\sigma')(s) + G_2(\sigma)(s)\,, \qquad (96) \\
K^{1,1}_{\gamma,\mathrm{e}}(\sigma)(s) &= K^{1,1}_{\eta,\mathrm{e}}(\sigma)(s) + G_2(\sigma)(s) \\
&= -\pi\, H(\sigma')(s) + G_2(\sigma)(s)\,, \qquad (97) \\
(c) \qquad K^{0,2}_{\gamma,\mathrm{i}}(\sigma)(s) &= -\left(\pi\, c(s) - \frac{2\,\pi^2}{L}\right)\sigma(s) + K^{0,2}_{\eta,\mathrm{i}}(\sigma)(s) + M_2^{\star}(\sigma)(s) \\
&= -\pi\, c(s)\,\sigma(s) + \pi\, H(\sigma')(s) + N_2^{\star}(\sigma)(s)\,, \qquad (98) \\
K^{0,2}_{\gamma,\mathrm{e}}(\sigma)(s) &= \left(\pi\, c(s) - \frac{2\,\pi^2}{L}\right)\sigma(s) + K^{0,2}_{\eta,\mathrm{e}}(\sigma)(s) + M_2^{\star}(\sigma)(s) \\
&= \pi\, c(s)\,\sigma(s) + \pi\, H(\sigma')(s) + N_2^{\star}(\sigma)(s)\,, \qquad (99)
\end{aligned}
$$

where $c(s)$ *denotes the curvature of* γ *at* $\gamma(s)$, *and* M_2, N_2, $G_2 : c[0,L] \to c[0,L]$ *are integral operators with kernels* $m_2(s,t)$, $n_2(s,t)$, $g_2(s,t) \in c^{k-2}([0,L]\times[0,L])$, *respectively. Furthermore,* $M_2^\star$, $N_2^\star$ *are the adjoints of* M_2, N_2, *the operator* G_2 *is self-adjoint, and* H *denotes the Hilbert transform (see (130) in Section 3.3).*

Remark 2.9 *The formulae (90) – (93) above are somewhat misleading, in that they state very simple facts in a relatively complicated manner. Specifically, each of the operators* $K^{1,0}_{\gamma,\mathrm{i}}$, $K^{1,0}_{\gamma,\mathrm{e}}$, $K^{0,1}_{\gamma,\mathrm{i}}$, $K^{0,1}_{\gamma,\mathrm{e}}$ *is a second kind integral operator with smooth* (c^{k-2}) *kernel (see, for example, [13]). In the case of the circle, the kernels of the operators* $K^{1,0}_{\eta,\mathrm{i}}$, $K^{1,0}_{\eta,\mathrm{e}}$, $K^{0,1}_{\eta,\mathrm{i}}$, $K^{0,1}_{\eta,\mathrm{e}}$ *are identically equal to* $-\frac{1}{2\pi}$. *Thus, (90) – (93) state the trivial fact that the difference of two smooth kernels is smooth. We list (90) – (93) for compatibility with the formulae (89), (94) – (99).*

Observation 2.10 *Formulae (89) – (99) have a straightforward interpretation. Specifically, each of the operators* K^0_γ, $K^{1,0}_{\gamma,\mathrm{i}}$, $K^{1,0}_{\gamma,\mathrm{e}}$, $K^{0,1}_{\gamma,\mathrm{i}}$, $K^{0,1}_{\gamma,\mathrm{e}}$, $K^{2,0}_{\gamma,\mathrm{i}}$, $K^{2,0}_{\gamma,\mathrm{e}}$, $K^{1,1}_{\gamma,\mathrm{i}}$, $K^{1,1}_{\gamma,\mathrm{e}}$, $K^{0,2}_{\gamma,\mathrm{i}}$, $K^{0,2}_{\gamma,\mathrm{e}}$, *is a sum of a standard operator (the corresponding operator on the circle) and an integral operator with a smooth kernel.*

In Section 4 below, a proof of formulae (94) and (95) is given; the proofs of the formulae (94) – (99) in Theorem 2.8 are similar and are omitted. Theorem 2.11 below extends the results of Theorem 2.8 above to the operators $K^{3,0}_{\gamma,\mathrm{i}}$, $K^{3,0}_{\gamma,\mathrm{e}}$, $K^{2,1}_{\gamma,\mathrm{i}}$, $K^{2,1}_{\gamma,\mathrm{e}}$, $K^{1,2}_{\gamma,\mathrm{i}}$, $K^{1,2}_{\gamma,\mathrm{e}}$, $K^{0,3}_{\gamma,\mathrm{i}}$, $K^{0,3}_{\gamma,\mathrm{e}}$. Its proof is virtually identical to that of Theorem 2.8, and is omitted.

Theorem 2.11 *Suppose that* $\gamma : [0,L] \to \mathbb{R}^2$ *is a* k *times continuously differentiable Jordan curve parametrized by its arclength, and that* $\eta : [0,L] \to \mathbb{R}^2$ *denotes the circle of radius* $\frac{L}{2\pi}$, *also parametrized by its arclength. Then, for any sufficiently smooth function* $\sigma : [0,L] \to \mathbb{R}$,

$$\begin{aligned}
(a)\qquad K^{3,0}_{\gamma,\mathrm{i}}(\sigma)(s) &= -\Big(2\pi\,\big(c(s)\big)^2 - \frac{4\pi^2}{L}\,c(s)\Big)\sigma(s) + \Big(\pi - \frac{L}{2}\,c(s)\Big)\,\sigma''(s) \\
&\quad -2\pi\,c'(s)\,H(\sigma)(s) + \frac{L}{2\pi}\,c(s)\,K^{3,0}_{\eta,\mathrm{i}}(\sigma)(s) + M_3(\sigma)(s) \\
&= -2\pi\,\big(c(s)\big)^2\sigma(s) + \pi\,\sigma''(s) - 2\pi\,c'(s)\,H(\sigma)(s) - 3\pi\,c(s)\,H(\sigma')(s) \\
&\quad +N_3(\sigma)(s), \qquad (100)
\end{aligned}$$

$$\begin{aligned}
K^{3,0}_{\gamma,\mathrm{e}}(\sigma)(s) &= \Big(2\pi\,\big(c(s)\big)^2 - \frac{4\pi^2}{L}\,c(s)\Big)\sigma(s) - \Big(\pi - \frac{L}{2}\,c(s)\Big)\,\sigma''(s) \\
&\quad -2\pi\,c'(s)\,H(\sigma)(s) + \frac{L}{2\pi}\,c(s)\,K^{3,0}_{\eta,\mathrm{e}}(\sigma)(s) + M_3(\sigma)(s) \\
&= 2\pi\,\big(c(s)\big)^2\sigma(s) - \pi\,\sigma''(s) - 2\pi\,c'(s)\,H(\sigma)(s) - 3\pi\,c(s)\,H(\sigma')(s) \\
&\quad +N_3(\sigma)(s), \qquad (101)
\end{aligned}$$

$$(b)\qquad K^{2,1}_{\gamma,\mathrm{i}}(\sigma)(s) = -\Big(\pi - \frac{L}{2}\,c(s)\Big)\,\sigma''(s) + \pi\,c'(s)\,H(\sigma)(s) + \frac{L}{2\pi}\,c(s)\,K^{2,1}_{\eta,\mathrm{i}}(\sigma)(s)$$

$$\begin{aligned}
& && +G_3(\sigma)(s) \\
& = && -\pi\,\sigma''(s) + \pi\, c'(s)\, H(\sigma)(s) + \pi\, c(s)\, H(\sigma')(s) + G_3(\sigma)(s)\,, && (102) \\
& K^{2,1}_{\gamma,\mathrm{e}}(\sigma)(s) = && \left(\pi - \frac{L}{2}\, c(s)\right) \sigma''(s) + \pi\, c'(s)\, H(\sigma)(s) + \frac{L}{2\,\pi}\, c(s)\, K^{2,1}_{\eta,\mathrm{e}}(\sigma)(s) \\
& && +G_3(\sigma)(s) \\
& = && \pi\,\sigma''(s) + \pi\, c'(s)\, H(\sigma)(s) + \pi\, c(s)\, H(\sigma')(s) + G_3(\sigma)(s)\,, && (103) \\
\textit{(c)} \quad & K^{1,2}_{\gamma,\mathrm{i}}(\sigma)(s) = && \left(\pi - \frac{L}{2}\, c(s)\right) \sigma''(s) + \frac{L}{2\,\pi}\, c(s)\, K^{1,2}_{\eta,\mathrm{i}}(\sigma)(s) + G^\star_3(\sigma)(s) \\
& = && \pi\,\sigma''(s) + \pi\, c(s)\, H(\sigma')(s) + G_3(\sigma)(s)\,, && (104) \\
& K^{1,2}_{\gamma,\mathrm{e}}(\sigma)(s) = && -\left(\pi - \frac{L}{2}\, c(s)\right) \sigma''(s) + \frac{L}{2\,\pi}\, c(s)\, K^{1,2}_{\eta,\mathrm{e}}(\sigma)(s) + G^\star_3(\sigma)(s) \\
& = && -\pi\,\sigma''(s) + \pi\, c(s)\, H(\sigma')(s) + G_3(\sigma)(s)\,, && (105) \\
\textit{(d)} \quad & K^{0,3}_{\gamma,\mathrm{i}}(\sigma)(s) = && \left(2\,\pi\left(c(s)\right)^2 - \frac{4\,\pi^2}{L}\, c(s)\right)\sigma(s) - \left(\pi - \frac{L}{2}\, c(s)\right) \sigma''(s) \\
& && -\pi\, c'(s)\, H(\sigma)(s) + \frac{L}{2\,\pi}\, c(s)\, K^{0,3}_{\eta,\mathrm{i}}(\sigma)(s) + M^\star_3(\sigma)(s) \\
& = && 2\,\pi\left(c(s)\right)^2\sigma(s) - \pi\,\sigma''(s) - \pi\, c'(s)\, H(\sigma)(s) - 3\,\pi\, c(s)\, H(\sigma')(s) \\
& && +N^\star_3(\sigma)(s)\,, && (106) \\
& K^{0,3}_{\gamma,\mathrm{e}}(\sigma)(s) = && -\left(2\,\pi\left(c(s)\right)^2 - \frac{4\,\pi^2}{L}\, c(s)\right)\sigma(s) + \left(\pi - \frac{L}{2}\, c(s)\right) \sigma''(s) \\
& && -\pi\, c'(s)\, H(\sigma)(s) + \frac{L}{2\,\pi}\, c(s)\, K^{0,3}_{\eta,\mathrm{e}}(\sigma)(s) + M^\star_3(\sigma)(s) \\
& = && -2\,\pi\left(c(s)\right)^2\sigma(s) + \pi\,\sigma''(s) - \pi\, c'(s)\, H(\sigma)(s) - 3\,\pi\, c(s)\, H(\sigma')(s) \\
& && +N^\star_3(\sigma)(s)\,, && (107)
\end{aligned}$$

where $c(s)$ *denotes the curvature of* γ *at* $\gamma(s)$*, and* M_3*,* N_3*,* $G_3 : c[0,L] \rightarrow c[0,L]$ *are integral operators with kernels* $m_3(s,t)$*,* $n_3(s,t)$*,* $g_3(s,t) \in c^{k-4}([0,L]\times[0,L])$*, respectively. Furthermore,* $M^\star_3$*,* N_3*,* $G^\star_3$ *are the adjoints of* M_3*,* N_3*,* G_3*, and* H *denotes the Hilbert transform (see (130) in Section 3.3).*

2.4 Computational Observations

In the numerical solution of elliptic PDEs, one is often confronted with the task of evaluating some (or all) of the operators (30) – (48) numerically. While this class of issues will be discussed in detail in a sequel to this paper, here we observe that an inspection of the formulae (50) – (68), (89) – (93), (94) – (99), (100) – (107) immediately shows that each of the operators (30) – (48) is a combination of the following: integral operators with smooth kernels, integral operators with the logarithmic singularity on the diagonal, the Hilbert transform, the derivative of the Hilbert transform, and the second derivative. The techniques for the accurate integration of smooth functions have been available for hundreds of years, and the numerical evaluation of the second derivative presents no serious problems. Effective techniques for the numerical evaluation of the Hilbert transform are less well-known, but have also been available for many years (see, for example, [16]). Efficient integration of logarithmically singular functions is also not very difficult (see [15, 8, 2]). The only possible source of problems is the derivative of the Hilbert transform; quadrature rules for the evaluation of the latter have been constructed, and will be published in [10]. Thus, there exist rapidly convergent schemes for the numerical evaluation of all of the operators (30) – (48), and, therefore, for the discretization of any problem of mathematical physics that has been reduced to a set of integro-pseudodifferential equations involving any (or all) of the operators (30) – (48).

Of course, when a problem of mathematical physics is discretized, one of principal issues is the condition number of the obtained system of equations. An examination of the formulae (51), (57), (52), (58) immediately shows that the operators $K^{1,0}_{\gamma,\mathrm{i}}$, $K^{0,1}_{\gamma,\mathrm{i}}$, $K^{1,0}_{\gamma,\mathrm{e}}$, $K^{0,1}_{\gamma,\mathrm{e}}$ are asymptotically well-conditioned (being a sum of the identity and a compact operator). The spectrum of the operator K^0_γ decays as $1/k$ with k the sequence number of the eigenvalue (see (50)), and its n-point discretization will (asymptotically) have condition number $\sim n$. Each of the operators $K^{2,0}_{\gamma,\mathrm{i}}$, $K^{1,1}_{\gamma,\mathrm{i}}$, $K^{0,2}_{\gamma,\mathrm{i}}$, $K^{2,0}_{\gamma,\mathrm{e}}$, $K^{1,1}_{\gamma,\mathrm{e}}$, $K^{0,2}_{\gamma,\mathrm{e}}$ has a spectrum that grows linearly, and the n-point discretization of each of them will have condition number n. Finally, each of the operators $K^{3,0}_{\gamma,\mathrm{i}}$, $K^{2,1}_{\gamma,\mathrm{i}}$, $K^{1,2}_{\gamma,\mathrm{i}}$, $K^{0,3}_{\gamma,\mathrm{i}}$, $K^{3,0}_{\gamma,\mathrm{e}}$, $K^{2,1}_{\gamma,\mathrm{e}}$, $K^{1,2}_{\gamma,\mathrm{e}}$, $K^{0,3}_{\gamma,\mathrm{e}}$ has a spectrum that grows as k^2; an n-point discretization of any of them will have condition number $\sim n^2$. Thus, whenever the problem to be solved results in the discretization of any one of the operators K^0_γ, $K^{2,0}_{\gamma,\mathrm{i}}$, $K^{1,1}_{\gamma,\mathrm{i}}$, $K^{0,2}_{\gamma,\mathrm{i}}$, $K^{2,0}_{\gamma,\mathrm{e}}$, $K^{1,1}_{\gamma,\mathrm{e}}$, $K^{0,2}_{\gamma,\mathrm{e}}$, $K^{3,0}_{\gamma,\mathrm{i}}$, $K^{2,1}_{\gamma,\mathrm{i}}$, $K^{1,2}_{\gamma,\mathrm{i}}$, $K^{0,3}_{\gamma,\mathrm{i}}$, $K^{3,0}_{\gamma,\mathrm{e}}$, $K^{2,1}_{\gamma,\mathrm{e}}$, $K^{1,2}_{\gamma,\mathrm{e}}$, $K^{0,3}_{\gamma,\mathrm{e}}$ there is a potential for condition number problems, similar to those encountered with direct discretization of differential equations.

Fortunately, formulae (50) – (68) suggest a solution. Specifically, an examination of the formulae (50), (53), (89), (94) immediately indicates that each of the operators $K^0_\gamma \circ K^{2,0}_{\gamma,\mathrm{i}}$, $K^{2,0}_{\gamma,\mathrm{i}} \circ K^0_\gamma$ is a sum of multiplication by a constant with a compact operator, i.e.

$$K^0_\gamma \circ K^{2,0}_{\gamma,\mathrm{i}} = \pi^2 \cdot I + M^{00,20}_\mathrm{i}, \tag{108}$$

$$K^{2,0}_{\gamma,\mathrm{i}} \circ K^0_\gamma = \pi^2 \cdot I + M^{20,00}_\mathrm{i}, \tag{109}$$

with $M^{20,00}_\mathrm{i}$, $M^{00,20}_\mathrm{i}$ compact operators $L^2[0,L] \to L^2[0,L]$. Similarly,

$$K^0_\gamma \circ K^{2,0}_{\gamma,\mathrm{e}} = \pi^2 \cdot I + M^{00,20}_\mathrm{e}, \tag{110}$$

$$K^{2,0}_{\gamma,\mathrm{e}} \circ K^0_\gamma = \pi^2 \cdot I + M^{20,00}_\mathrm{e}, \tag{111}$$

and

$$K^0_\gamma \circ K^{1,1}_{\gamma,\mathrm{i}} = -\pi^2 \cdot I + M^{00,11}_\mathrm{i}, \tag{112}$$

$$K^{1,1}_{\gamma,\mathrm{i}} \circ K^0_\gamma = -\pi^2 \cdot I + M^{11,00}_\mathrm{i}, \tag{113}$$

$$K^0_\gamma \circ K^{1,1}_{\gamma,\mathrm{e}} = -\pi^2 \cdot I + M^{00,11}_\mathrm{e}, \tag{114}$$

$$K^{1,1}_{\gamma,\mathrm{e}} \circ K^0_\gamma = -\pi^2 \cdot I + M^{11,00}_{\mathrm{e}}, \tag{115}$$

and

$$K^0_\gamma \circ K^{0,2}_{\gamma,\mathrm{i}} = \pi^2 \cdot I + M^{00,02}_{\mathrm{i}}, \tag{116}$$

$$K^{0,2}_{\gamma,\mathrm{i}} \circ K^0_\gamma = \pi^2 \cdot I + M^{02,00}_{\mathrm{i}}, \tag{117}$$

$$K^0_\gamma \circ K^{0,2}_{\gamma,\mathrm{e}} = \pi^2 \cdot I + M^{00,02}_{\mathrm{e}}, \tag{118}$$

$$K^{0,2}_{\gamma,\mathrm{e}} \circ K^0_\gamma = \pi^2 \cdot I + M^{02,00}_{\mathrm{e}}; \tag{119}$$

all of the operators $M^{11,00}_{\mathrm{i}}$, $M^{11,00}_{\mathrm{e}}$, $M^{00,11}_{\mathrm{i}}$, $M^{00,11}_{\mathrm{e}}$, $M^{02,00}_{\mathrm{i}}$, $M^{02,00}_{\mathrm{e}}$, $M^{00,02}_{\mathrm{i}}$, $M^{00,02}_{\mathrm{e}}$ are compact. In other words, the operator K^0_γ is a perfect preconditioner (asymptotically speaking) for each of the second order pseudodifferential operators of potential theory in two dimensions; in turn, K^0_γ is preconditioned by each of the operators (94) – (99).

Expressions (100) – (107) contain the second derivative, and are, clearly, preconditioned by the operator of repeated integration $I_2 : L^2[0,L] \to L^2[0,L]$, defined by its action on the functions $e^{i\cdot m\cdot x/L}$ via the formula

$$I_2(e^{i\cdot m\cdot x/L}) = \frac{1}{m^2} \cdot e^{i\cdot m\cdot x/L}. \tag{120}$$

In other words, for each of the operators (30) – (48), there is available a straightforward preconditioner. Numerical implications of these (and related) observations will be discussed in [10].

3 Analytical Preliminaries

3.1 Principal Value Integrals

Integrals of the form

$$\int_a^b \frac{\varphi(t)}{t-s}\,dt, \tag{121}$$

where $s \in (a,b)$, do not exist in the classical sense, and are often referred to as *singular integrals.*

Definition 3.1 *Suppose that φ is a function $[a,b] \to \mathbb{R}$, $s \in (a,b)$, and the limit*

$$\lim_{\epsilon\to 0}\left(\int_a^{s-\epsilon} \frac{\varphi(t)}{t-s}\,dt + \int_{s+\epsilon}^b \frac{\varphi(t)}{t-s}\,dt\right) \tag{122}$$

exists and is finite. Then we will denote the limit (122) by

$$\text{p.v.} \int_a^b \frac{\varphi(t)}{t-s}\,dt, \tag{123}$$

and refer to it as a principal value integral.

Theorem 3.1 *Suppose that the function $\varphi : [a,b] \to \mathbb{R}$ is continuously differentiable in a neighborhood of $s \in (a,b)$. Then the principal value integral (123) exists.*

3.2 Finite Part Integrals

In this paper, we will be dealing with integrals of the form

$$\int_a^b \frac{\varphi(t)}{(t-s)^2}\,dt\,, \tag{124}$$

where $s \in (a,b)$, which are divergent in the classical sense. This type of integrals are often referred to as *hypersingular* or *strongly singular.*

Definition 3.2 *Suppose that φ is a function $[a,b] \to \mathbb{R}$, $s \in (a,b)$, and the limit*

$$\lim_{\epsilon \to 0}\left(\int_a^{s-\epsilon} \frac{\varphi(t)}{(t-s)^2}\,dt + \int_{s+\epsilon}^b \frac{\varphi(t)}{(t-s)^2}\,dt - \frac{2\varphi(s)}{\epsilon}\right) \tag{125}$$

exists and is finite. Then we will denote the limit (125) by

$$\text{f.p.}\int_a^b \frac{\varphi(t)}{(t-s)^2}\,dt\,, \tag{126}$$

and refer to it as a finite part integral (see, for example, [7]).

The following obvious theorem provides sufficient conditions for the existence of the finite part integral (125), and establishes a connection between finite part and principal value integrals.

Theorem 3.2 *Suppose that the function $\varphi : [a,b] \to \mathbb{R}$ is twice continuously differentiable in a neighborhood of $s \in (a,b)$. Then the finite part integral (126) exists, and*

$$\text{f.p.}\int_a^b \frac{\varphi(t)}{(t-s)^2}\,dt = \frac{d}{ds}\,\text{p.v.}\int_a^b \frac{\varphi(t)}{t-s}\,dt\,. \tag{127}$$

3.3 The Hilbert Transform

For an arbitrary periodic function $\varphi \in L^2[-\pi,\pi]$ and any integer k, we will denote by $\widehat{\varphi}_k$ the k-th Fourier coefficient of φ, defined by the formula,

$$\widehat{\varphi}_k = \frac{1}{2\pi}\int_{-\pi}^{\pi} \varphi(s)\, e^{-iks}\,ds\,, \tag{128}$$

so that

$$\varphi(t) = \sum_{k=-\infty}^{\infty} \widehat{\varphi}_k\, e^{ikt}\,, \tag{129}$$

for all $t \in [-\pi,\pi]$.

Definition 3.3 *The Hilbert transform is the mapping $H : L^2[-\pi,\pi] \to L^2[-\pi,\pi]$, given by the formula*

$$H(\varphi)(s) = \sum_{\substack{k=-\infty \\ k\neq 0}}^{\infty} -i \, \mathrm{sgn}(k) \, \hat{\varphi}_k \, e^{iks} \,, \tag{130}$$

with $\varphi \in L^2[-\pi,\pi]$ an arbitrary function. The function $H(\varphi) : [-\pi,\pi] \to \mathbb{C}$ is often referred to as the conjugate function of φ.

The following theorem summarizes several well-known properties of the Hilbert transform (see, for example, [9]).

Theorem 3.3 *(a) The mapping $H : L^2[-\pi,\pi] \to L^2[-\pi,\pi]$ is bounded.*
(b) For any integrable φ, the identity

$$H(\varphi)(s) = \text{p.v.} \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{\varphi(t)}{\tan\left(\frac{s-t}{2}\right)} \, dt \,, \tag{131}$$

holds almost everywhere.
(c) For any function $\varphi \in c^1[-\pi,\pi]$,

$$H(\varphi')(s) = \left((H(\varphi))' \right)(s) = \sum_{\substack{k=-\infty \\ k\neq 0}}^{\infty} |k| \, \hat{\varphi}_k \, e^{iks} \,. \tag{132}$$

In other words,

$$HD = DH \,, \tag{133}$$

where $D = \frac{d}{ds}$ is the differentiation operator.

3.4 Boundary Integral Operators

In this subsection, we define boundary the integral operators $K^{1,0}_\gamma$, $K^{2,0}_\gamma$, $K^{3,0}_\gamma$, $K^{0,1}_\gamma$, $K^{1,1}_\gamma$, $K^{2,1}_\gamma$, $K^{0,2}_\gamma$, $K^{1,2}_\gamma$, $K^{0,3}_\gamma$, that are closely related to the operators (31) – (48) defined in Section 2.

Definition 3.4 *Suppose that the function $\sigma : [0,L] \to \mathbb{R}$ is sufficiently smooth. Then we denote by $K^{1,0}_\gamma, K^{0,1}_\gamma : c[0,L] \to c[0,L]$ and $K^{2,0}_\gamma, K^{3,0}_\gamma, K^{1,1}_\gamma, K^{2,1}_\gamma, K^{0,2}_\gamma, K^{1,2}_\gamma, K^{0,3}_\gamma : c^2[0,L] \to c[0,L]$ the operators defined by the formulae*

$$K^{1,0}_\gamma(\sigma)(s) = \int_0^L \frac{\partial \Phi_{\gamma(t)}(\gamma(s))}{\partial N(t)} \sigma(t) \, dt \,, \tag{134}$$

$$K^{2,0}_\gamma(\sigma)(s) = \text{f.p.} \int_0^L \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s))}{\partial N(t)^2} \sigma(t) \, dt \,, \tag{135}$$

$$K^{3,0}_\gamma(\sigma)(s) = \text{f.p.} \int_0^L \frac{\partial^3 \Phi_{\gamma(t)}(\gamma(s))}{\partial N(t)^3} \sigma(t) \, dt \,, \tag{136}$$

$$K_\gamma^{0,1}(\sigma)(s) = \int_0^L \frac{\partial \Phi_{\gamma(t)}(\gamma(s))}{\partial N(s)}\,\sigma(t)\,dt\,, \tag{137}$$

$$K_\gamma^{1,1}(\sigma)(s) = \text{f.p.}\int_0^L \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s))}{\partial N(s)\,\partial N(t)}\,\sigma(t)\,dt\,, \tag{138}$$

$$K_\gamma^{2,1}(\sigma)(s) = \text{f.p.}\int_0^L \frac{\partial^3 \Phi_{\gamma(t)}(\gamma(s))}{\partial N(s)\,\partial N(t)^2}\,\sigma(t)\,dt\,, \tag{139}$$

$$K_\gamma^{0,2}(\sigma)(s) = \text{f.p.}\int_0^L \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s))}{\partial N(s)^2}\,\sigma(t)\,dt\,, \tag{140}$$

$$K_\gamma^{1,2}(\sigma)(s) = \text{f.p.}\int_0^L \frac{\partial^3 \Phi_{\gamma(t)}(\gamma(s))}{\partial N(s)^2\,\partial N(t)}\,\sigma(t)\,dt\,, \tag{141}$$

$$K_\gamma^{0,3}(\sigma)(s) = \text{f.p.}\int_0^L \frac{\partial^3 \Phi_{\gamma(t)}(\gamma(s))}{\partial N(s)^3}\,\sigma(t)\,dt\,, \tag{142}$$

respectively.

Remark 3.4 *Obviously, the operators $K_\gamma^{0,1}$, $K_\gamma^{0,2}$, $K_\gamma^{0,3}$, $K^{1,2}$ given by the formulae (137), (140) – (142) are the adjoints of the operators $K_\gamma^{1,0}$, $K_\gamma^{2,0}$, $K_\gamma^{3,0}$, $K_\gamma^{2,1}$ defined by (134) – (136), (139). Furthermore, $K_\gamma^{1,1}$, defined by (138), is self-adjoint.*

4 Proof of Results

In this section we prove the results from Section 2. The outline of this section is as follows: First, we consider the case where γ is a circle. We provide the proof for Theorem 2.6. In Lemma 4.2 we give explicit formulas for the boundary integral operators (134) – (140) for the case where γ is a circle. Then, by combining Theorem 2.6 and Lemma 4.2, we immediately get the so-called jump conditions for the operators (12) – (25) on a circle. These are stated in Theorem 4.3.

Next, we consider the case where γ is an arbitrary and sufficiently smooth Jordan curve. Since the proof of the identities (94) – (99) in Theorem 2.8 are similar, we only provide the proof for (94) and (95). In fact, (94) and (95) in Theorem 2.8 follow immediately from Theorem 4.7 and Lemma 4.6. The proof of Theorem 4.7 is based on Theorem 4.3 and the approximation (178) given in Lemma 4.5.

Proof of Theorem 2.6 Since the proofs for the identities (50) – (64) are nearly identical, we only provide the proof for the interior limit of the quadruple layer potential (53). Further, it is enough to prove (53) for the case $r = 1$; the general case follows by a simple transformation of variables. We choose the parametrization

$$\gamma(t) = (\cos(t), \sin(t))\,, \tag{143}$$

where $t \in [-\pi, \pi]$. It immediately follows from (143) that

$$\begin{aligned}
&\int_{-\pi}^{\pi} \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s) - h \cdot N(s))}{\partial N(t)^2} \, e^{ikt} \, dt = \\
&\quad = \int_{-\pi}^{\pi} \frac{1 - 2 \cdot (1-h) \cdot \cos(t-s) + (1-h)^2 \cdot \cos(2\,(t-s))}{(1 + (1-h)^2 - 2 \cdot (1-h) \cdot \cos(t-s))^2} \, e^{ikt} \, dt \\
&\quad = e^{iks} \cdot \int_{-\pi}^{\pi} \frac{1 - 2 \cdot (1-h) \cdot \cos(t) + (1-h)^2 \cdot \cos(2\,t)}{(1 + (1-h)^2 - 2 \cdot (1-h) \cdot \cos(t))^2} \, e^{ikt} \, dt\,,
\end{aligned} \tag{144}$$

for any $s \in [-\pi, \pi]$. We will use calculus of residues to evaluate the integral (144). To this effect, the substitution

$$z = e^{it}\,, \tag{145}$$

converts (144) into

$$\begin{aligned}
&e^{iks} \cdot \int_{-\pi}^{\pi} \frac{1 - 2 \cdot (1-h) \cdot \cos(t) + (1-h)^2 \cdot \cos(2\,t)}{(1 + (1-h)^2 - 2 \cdot (1-h) \cdot \cos(t))^2} \, e^{ikt} \, dt = \\
&\quad = e^{iks} \cdot \int_{|z|=1} \frac{-i}{z} \left(\frac{1 - (1-h)\,(z + z^{-1}) + \frac{1}{2}\,(1-h)^2\,(z^2 + z^{-2})}{(1 + (1-h)^2 - (1-h)\,(z + z^{-1}))^2} \right) \cdot z^k \, dz\,,
\end{aligned} \tag{146}$$

and after simple algebraic manipulation, we get

$$\begin{aligned}
&\frac{-i}{z} \left(\frac{1 - (1-h)\,(z + z^{-1}) + \frac{1}{2}\,(1-h)^2\,(z^2 + z^{-2})}{(1 + (1-h)^2 - (1-h)\,(z + z^{-1}))^2} \right) \cdot z^k = \\
&\quad = \frac{1}{2} \cdot \left(-\frac{i z^{k+1}}{((1-h) - z)^2} - \frac{i z^{k-1}}{(z\,(1-h) - 1)^2} \right)\,.
\end{aligned} \tag{147}$$

Substituting (147) into (146), we obtain

$$\begin{aligned}
&\int_{-\pi}^{\pi} \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s) - h \cdot N(s))}{\partial N(t)^2} \, e^{ikt} \, dt = \\
&\quad = e^{iks} \cdot \int_{|z|=1} \frac{1}{2} \cdot \left(-\frac{i z^{k+1}}{((1-h) - z)^2} - \frac{i z^{k-1}}{(z\,(1-h) - 1)^2} \right) dz\,.
\end{aligned} \tag{148}$$

Now, formula (53) for $r = 1$ follows by applying a standard residue calculation to (148). □

Remark 4.1 *Formulae (50) - (52), (57) - (58) follow from well-known results (see for example [11, 3]). While the derivation of (53) - (56), (59) - (64) is quite similar, the authors failed to find them in the literature.*

The operators $K_\gamma^{1,0}$, $K_\gamma^{2,0}$, $K_\gamma^{3,0}$, $K_\gamma^{1,1}$, $K_\gamma^{2,1}$, $K_\gamma^{0,1}$, $K_\gamma^{0,2}$, $K_\gamma^{0,3}$, $K_\gamma^{1,2}$ defined by (134) - (141), assume a particularly simple form on the circle. The following lemma follows immediately from an elementary computation.

Lemma 4.2 *Suppose that γ is a circle of radius r parametrized by its arclength with exterior unit normal denoted by N. Then, for any sufficiently smooth function $\sigma : [-\pi r, \pi r] \to \mathbb{C}$:*

$$\text{(a)} \qquad K_\gamma^{1,0}(\sigma)(s) = \int_{-\pi r}^{\pi r} -\frac{\sigma(t)}{2\,r}\,dt = -\pi\,\widehat{\sigma}_0\,, \tag{149}$$

$$\begin{aligned} \text{(b)} \qquad K_\gamma^{2,0}(\sigma)(s) &= \text{f.p.}\int_{-\pi r}^{\pi r}\left(\frac{1}{2\,r^2} + \frac{1}{2\,r^2\,\cos(\frac{t-s}{r}) - 2\,r^2}\right)\sigma(t)\,dt \\ &= \pi\,r^{-1}\,\widehat{\sigma}_0 + \pi\,H(\sigma')(s)\,, \end{aligned} \tag{150}$$

$$\begin{aligned} \text{(c)} \qquad K_\gamma^{3,0}(\sigma)(s) &= \text{f.p.}\int_{-\pi r}^{\pi r}\left(-\frac{1}{r^3} - \frac{3}{2\,r^3\,\cos(\frac{t-s}{r}) - 2\,r^3}\right)\sigma(t)\,dt \\ &= -2\,\pi\,r^{-2}\,\widehat{\sigma}_0 - 3\,\pi\,r^{-1}\,H(\sigma')(s)\,, \end{aligned} \tag{151}$$

$$\text{(d)} \qquad K_\gamma^{0,1}(\sigma)(s) = \int_{-\pi r}^{\pi r} -\frac{\sigma(t)}{2\,r}\,dt = -\pi\,\widehat{\sigma}_0\,, \tag{152}$$

$$\text{(e)} \qquad K_\gamma^{1,1}(\sigma)(s) = \text{f.p.}\int_{-\pi r}^{\pi r}\frac{\sigma(t)}{2\,r^2 - 2\,r^2\,\cos(\frac{t-s}{r})}\,dt = -\pi\,H(\sigma')(s)\,, \tag{153}$$

$$\text{(f)} \qquad K_\gamma^{2,1}(\sigma)(s) = \text{f.p.}\int_{-\pi r}^{\pi r}\frac{\sigma(t)}{2\,r^3\,\cos(\frac{t-s}{r}) - 2\,r^3}\,dt = \pi\,r^{-1}\,H(\sigma')(s)\,, \tag{154}$$

$$\begin{aligned} \text{(g)} \qquad K_\gamma^{0,2}(\sigma)(s) &= \text{f.p.}\int_{-\pi r}^{\pi r}\left(\frac{1}{2\,r^2} + \frac{1}{2\,r^2\,\cos(\frac{t-s}{r}) - 2\,r^2}\right)\sigma(t)\,dt \\ &= \pi\,r^{-1}\,\widehat{\sigma}_0 + \pi\,H(\sigma')(s)\,, \end{aligned} \tag{155}$$

$$\text{(h)} \qquad K_\gamma^{1,2}(\sigma)(s) = \text{f.p.}\int_{-\pi r}^{\pi r}\frac{\sigma(t)}{2\,r^3\,\cos(\frac{t-s}{r}) - 2\,r^3}\,dt = \pi\,r^{-1}\,H(\sigma')(s)\,, \tag{156}$$

$$\begin{aligned} \text{(i)} \qquad K_\gamma^{0,3}(\sigma)(s) &= \text{f.p.}\int_{-\pi r}^{\pi r}\left(-\frac{1}{r^3} - \frac{3}{2\,r^3\,\cos(\frac{t-s}{r}) - 2\,r^3}\right)\sigma(t)\,dt \\ &= -2\,\pi\,r^{-2}\,\widehat{\sigma}_0 - 3\,\pi\,r^{-1}\,H(\sigma')(s)\,, \end{aligned} \tag{157}$$

where H denotes the Hilbert transform (see (130) in Section 3.3).

The following theorem is an immediate consequence of Theorem 2.6 and Lemma 4.2. It summarizes the so-called jump conditions for the integrals (12) – (29) on the boundary Γ, where Γ is a circle.

Theorem 4.3 *Suppose that γ is a circle of radius r parametrized by its arclength with exterior unit normal denoted by N. Further, suppose that H denotes the Hilbert transform (130). Then, for any sufficiently smooth function $\sigma : [-\pi r, \pi r] \to \mathbb{C}$,*

$$\text{(a)} \qquad K_{\gamma,\mathrm{i}}^{1,0}(\sigma)(s) = -\pi\,\sigma(s) + K_\gamma^{1,0}(\sigma)(s)\,, \tag{158}$$

$$K^{1,0}_{\gamma,\mathrm{e}}(\sigma)(s) = \pi\,\sigma(s) + K^{1,0}_{\gamma}(\sigma)(s)\,, \tag{159}$$

(b)

$$K^{2,0}_{\gamma,\mathrm{i}}(\sigma)(s) = \pi\,r^{-1}\,\sigma(s) + K^{2,0}_{\gamma}(\sigma)(s)\,, \tag{160}$$

$$K^{2,0}_{\gamma,\mathrm{e}}(\sigma)(s) = -\pi\,r^{-1}\,\sigma(s) + K^{2,0}_{\gamma}(\sigma)(s)\,, \tag{161}$$

(c)

$$K^{3,0}_{\gamma,\mathrm{i}}(\sigma)(s) = -2\,\pi\,r^{-2}\,\sigma(s) + \pi\,\sigma''(s) + K^{3,0}_{\gamma}(\sigma)(s)\,, \tag{162}$$

$$K^{3,0}_{\gamma,\mathrm{e}}(\sigma)(s) = 2\,\pi\,r^{-2}\,\sigma(s) - \pi\,\sigma''(s) + K^{3,0}_{\gamma}(\sigma)(s)\,, \tag{163}$$

(d)

$$K^{0,1}_{\gamma,\mathrm{i}}(\sigma)(s) = \pi\,\sigma(s) + (K^{1,0}_{\gamma})^{\star}(\sigma)(s)\,, \tag{164}$$

$$K^{0,1}_{\gamma,\mathrm{e}}(\sigma)(s) = -\pi\,\sigma(s) + (K^{1,0}_{\gamma})^{\star}(\sigma)(s)\,, \tag{165}$$

(e)

$$K^{1,1}_{\gamma,\mathrm{i}}(\sigma)(s) = K^{1,1}_{\gamma,\mathrm{e}}(\sigma)(s) = K^{1,1}_{\gamma}(\sigma)(s) = -\pi\,H(\sigma')(s)\,, \tag{166}$$

(f)

$$K^{2,1}_{\gamma,\mathrm{i}}(\sigma)(s) = -\pi\,\sigma''(s) + K^{2,1}_{\gamma}(\sigma)(s)\,, \tag{167}$$

$$K^{2,1}_{\gamma,\mathrm{e}}(\sigma)(s) = \pi\,\sigma''(s) + K^{2,1}_{\gamma}(\sigma)(s)\,, \tag{168}$$

(g)

$$K^{0,2}_{\gamma,\mathrm{i}}(\sigma)(s) = -\pi\,r^{-1}\,\sigma(s) + K^{0,2}_{\gamma}(\sigma)(s)\,, \tag{169}$$

$$K^{0,2}_{\gamma,\mathrm{e}}(\sigma)(s) = \pi\,r^{-1}\,\sigma(s) + K^{0,2}_{\gamma}(\sigma)(s)\,, \tag{170}$$

(h)

$$K^{1,2}_{\gamma,\mathrm{i}}(\sigma)(s) = \pi\,\sigma''(s) + (K^{2,1}_{\gamma})^{\star}(\sigma)(s)\,, \tag{171}$$

$$K^{1,2}_{\gamma,\mathrm{e}}(\sigma)(s) = -\pi\,\sigma''(s) + (K^{2,1}_{\gamma})^{\star}(\sigma)(s)\,, \tag{172}$$

(i)

$$K^{0,3}_{\gamma,\mathrm{i}}(\sigma)(s) = 2\,\pi\,r^{-2}\,\sigma(s) - \pi\,\sigma''(s) + (K^{3,0}_{\gamma})^{\star}(\sigma)(s)\,, \tag{173}$$

$$K^{0,3}_{\gamma,\mathrm{e}}(\sigma)(s) = -2\,\pi\,r^{-2}\,\sigma(s) + \pi\,\sigma''(s) + (K^{3,0}_{\gamma})^{\star}(\sigma)(s)\,. \tag{174}$$

We now proceed to the case where γ is an arbitrary sufficiently smooth Jordan curve. The following obvious lemma can be found in most elementary textbooks on differential geometry (see, for example, [4]).

Lemma 4.4 *Suppose that $\gamma : [0, L] \to \mathbb{R}^2$ is a sufficiently smooth Jordan curve parametrized by its arclength with the exterior unit normal and the unit tangent vectors at $\gamma(s)$ denoted by $N(s)$ and $T(s)$, respectively. Then, there exist a positive real number a (dependent on γ), and two continuously differentiable functions $f, g : (-a, a) \to \mathbb{R}$ (dependent on γ), such that for any $s \in [0, L]$,*

$$\gamma(s+t) - \gamma(s) = \Big(t + t^3 \cdot f(t)\Big) \cdot T(s) - \left(\frac{c\,t^2}{2} + t^3 \cdot g(t)\right) \cdot N(s)\,, \tag{175}$$

for all $t \in (-a, a)$, where the coefficient c in (175) is the curvature of γ at the point $\gamma(s)$. Furthermore, for all $t \in (-a, a)$,

$$|f(t)| \le \|\gamma'''(s)\|\,, \tag{176}$$

$$|g(t)| \le \|\gamma'''(s)\|\,. \tag{177}$$

In the local parametrization (175), the potential of a quadrupole located at $\gamma(s)$ and oriented in the direction $N(s)$ assumes a particularly simple form, given by the following lemma.

Lemma 4.5 *Suppose that $\gamma : [0, L] \to \mathbb{R}^2$ is a sufficiently smooth Jordan curve parametrized by its arclength. Then, there exist real positive numbers A, a and h_0 such that for any $s \in [0, L]$*

$$\left| \frac{\partial^2 \Phi_{\gamma(s+t)}(\gamma(s) - h \cdot N(s))}{\partial N(s+t)^2} - \frac{h^2 - t^2}{(h^2+t^2)^2} - \frac{c\,h\,t^2\,(5\,h^2 + t^2)}{(h^2+t^2)^3} \right| \le A\,, \tag{178}$$

for all $t \in (-a, a)$, $0 \le h < h_0$, where the coefficient c in (178) is the curvature of γ at the point $\gamma(s)$.

Proof. Without loss of generality, it is sufficient to prove the lemma for the case where $s = 0$, $\gamma(0) = 0$, and $\gamma'(0) = (1, 0)$. Substituting (175) into (9) and evaluating the result at $x = (0, h)$, we obtain

$$\frac{\partial^2 \Phi_{\gamma(t)}(x)}{\partial N(t)^2} \;=\; \frac{p_0(h,t)}{(h^2 + t^2 + r(h,t))^2}\,, \tag{179}$$

where $p_0, r : \mathbb{R}^2 \to \mathbb{R}$ are functions given by the formulae

$$\begin{aligned}
p_0(h,t) \;=\; & \Big[h - t + c\,h\,t + \frac{c\,t^2}{2} - \frac{c^2\,t^3}{2} + 3\,h\,t^2\big(f(t) + g(t)\big) - 2\,t^3\big(2\,f(t) - g(t)\big) \\
& - \frac{c\,t^4}{2}\big(f(t) + 5\,g(t)\big) + h\,t^3\big(f'(t) + g'(t)\big) - t^4\big(f'(t) - g'(t)\big) - 3\,t^5\big(f(t)^2 + g(t)^2\big) \\
& - \frac{c\,t^5}{2}\big(f'(t) + g'(t)\big) - t^6\,f(t)\big(f'(t) - g'(t)\big) - t^6\,g(t)\big(f'(t) + g'(t)\big) \Big] \\
& \cdot \Big[h + t - c\,h\,t + \frac{c\,t^2}{2} + \frac{c^2\,t^3}{2} + 3\,h\,t^2\big(f(t) - g(t)\big) + 2\,t^3\big(2\,f(t) + g(t)\big) \\
& - \frac{c\,t^4}{2}\big(f(t) - 5\,g(t)\big) + h\,t^3\big(f'(t) - g'(t)\big) + t^4\big(f'(t) + g'(t)\big) + 3\,t^5\big(f(t)^2 + g(t)^2\big) \\
& - \frac{c\,t^5}{2}\big(f'(t) - g'(t)\big) + t^6\,f(t)\big(f'(t) + g'(t)\big) - t^6\,g(t)\big(f'(t) - g'(t)\big) \Big]\,,
\end{aligned} \tag{180}$$

$$r(h,t) \;=\; -c\,h\,t^2 - 2\,h\,t^3\,g(t) + \frac{c^2\,t^4}{4} + 2\,t^4\,f(t) + c\,t^5\,g(t) + t^6\big(f(t)^2 + g(t)^2\big)\,. \tag{181}$$

We also introduce the notation

$$p_1(h,t) = \big(h^2 + t^2 + r(h,t)\big)^2 - \big(h^2 + t^2\big)^2 = 2\,(h^2 + t^2) \cdot r(h,t) + r(h,t)^2\,. \tag{182}$$

Obviously, (180) – (182) are algebraic combinations of f, g, f', g', t, and h, and an examination of formulae (180) – (182) immediately shows that there exist positive real numbers a, h_0, and

C (dependent on γ) such that

$$\begin{aligned} \left| p_0(h,t) - h^2 + t^2 - 3\,c\,h\,t^2 \right| &\le C(h^2+t^2)^2\,, &(183)\\ \left| p_0(h,t)\cdot p_1(h,t) - 2\,c\,h\,t^2\,(h^2+t^2)\,(h^2-t^2) \right| &\le C\,(h^2+t^2)^4\,, &(184)\\ \left| p_0(h,t)\cdot p_1(h,t)^2 \right| &\le C\,(h^2+t^2)^6\,, &(185)\\ \left| \frac{p_1(h,t)}{(h^2+t^2)^2} \right| &< 1\,, &(186) \end{aligned}$$

for all $h < h_0$, $t \in (-a,a)$. Substituting (182) into (179), we have

$$\begin{aligned} \frac{\partial^2 \Phi_{\gamma(t)}(x)}{\partial N(t)^2} &= \frac{p_0(h,t)}{(h^2+t^2)^2\Big(1+\frac{p_1(h,t)}{(h^2+t^2)^2}\Big)} \\ &= \frac{p_0(h,t)}{(h^2+t^2)^2} \sum_{k=0}^{\infty} (-1)^k \,\frac{p_1(h,t)^k}{(h^2+t^2)^{2k}}\,, \end{aligned} \tag{187}$$

where the convergence of the series follows from (186). Combining (183) – (185), we obtain

$$\begin{aligned} &\left| \frac{\partial^2 \Phi_{\gamma(t)}(x)}{\partial N(t)^2} - \frac{h^2-t^2}{(h^2+t^2)^2} - \frac{c\,h\,t^2\,(5\,h^2+t^2)}{(h^2+t^2)^3} \right| \le \left| \frac{p_0(h,t) - h^2 + t^2 - 3\,c\,h\,t^2}{(h^2+t^2)^2} \right| \\ &\quad + \left| \frac{p_0(h,t)\cdot p_1(h,t) - 2\,c\,h\,t^2\,(h^2+t^2)\,(h^2-t^2)}{(h^2+t^2)^4} \right| + \sum_{k=2}^{\infty} \left| \frac{p_0(h,t)\cdot p_1(h,t)^k}{(h^2+t^2)^{2k+2}} \right| \\ &\le 2\,C + C\cdot \frac{\alpha^2}{1-\alpha}\,, \end{aligned} \tag{188}$$

with α defined by the formula

$$\alpha = \sup_{h<h_0\,,\,t\in(-a,a)} \left| \frac{p_1(h,t)}{(h^2+t^2)^2} \right| . \tag{189}$$

Now, introducing the notation

$$A = 2\,C + C\cdot \frac{\alpha^2}{1-\alpha}\,, \tag{190}$$

we obtain (178). □

Lemma 4.2 provides an explicit formula for the operator $K_\gamma^{2,0}$, defined in (135), in the case when γ is a circle. The following lemma shows that the operator $K_\gamma^{2,0}$ on an arbitrary sufficiently smooth Jordan curve of length L, is a compact perturbation of $K_\gamma^{2,0}$ on the circle of radius $\frac{L}{2\pi}$. Its proof is an immediate consequence of estimate (178) in Lemma 4.5.

Lemma 4.6 *Suppose that $\gamma : [0,L] \to \mathbb{R}^2$ is a sufficiently smooth Jordan curve parametrized by its arclength, and that $\eta : [0,L] \to \mathbb{R}^2$ denotes the circle of radius $\frac{L}{2\pi}$, also parametrized* *by its arclength. In addition, suppose that $\sigma : [0, L] \to \mathbb{R}$ is a twice continuously differentiable function. Then,*

$$\text{f.p.} \int_0^L \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s))}{\partial N(t)^2}\, \sigma(t)\, dt = \text{f.p.} \int_0^L \frac{\partial^2 \Phi_{\eta(t)}(\eta(s))}{\partial N(t)^2}\, \sigma(t)\, dt + M_2(\sigma)(s)\,, \tag{191}$$

where $M_2 : c[0, L] \to c[0, L]$ is a compact operator defined by the formula

$$M_2(\sigma)(s) = \int_0^L \left(\frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s))}{\partial N(t)^2} - \frac{\partial^2 \Phi_{\eta(t)}(\eta(s))}{\partial N(t)^2} \right) \sigma(t)\, dt\,. \tag{192}$$

Furthermore, for any $t \neq s$,

$$\begin{aligned} m_2(s,t) \;\; &= \;\; \frac{2\,\langle N(t), \gamma(s) - \gamma(t)\rangle^2}{\|\gamma(s) - \gamma(t)\|^4} - \frac{1}{2} \left(\frac{2\,\pi}{L}\right)^2 \\ &\quad + \frac{\|\gamma(s) - \gamma(t)\|^2 - 2\left(\frac{L}{2\,\pi}\right)^2 \left(1 - \cos\left(\frac{2\pi}{L}(s-t)\right)\right)}{\|\gamma(s) - \gamma(t)\|^2\, 2\left(\frac{L}{2\,\pi}\right)^2 \left(1 - \cos\left(\frac{2\pi}{L}(s-t)\right)\right)}\,, \end{aligned} \tag{193}$$

and for $t = s$,

$$m_2(s,s) = \frac{5}{12}\left(c(s)\right)^2 - \frac{5}{12}\left(\frac{2\,\pi}{L}\right)^2, \tag{194}$$

where $c(s)$ is the curvature of γ at the point $\gamma(s)$, and $m_2 : [0, L] \times [0, L] \to \mathbb{R}$ is the kernel of the operator M_2.

The following theorem provides the so-called jump conditions for the operators (14) and (15) on the boundary Γ, when Γ is sufficiently smooth.

Theorem 4.7 *Suppose that $\gamma : [0, L] \to \mathbb{R}^2$ is a sufficiently smooth Jordan curve parametrized by its arclength. Then, for any sufficiently smooth function $\sigma : [0, L] \to \mathbb{R}$,*

$$\begin{aligned} &K^{2,0}_{\gamma,\mathrm{e}}(\sigma)(s) - K^{2,0}_{\gamma,\mathrm{i}}(\sigma)(s) = \\ &\quad = \lim_{h \to 0} \int_0^L \left(\frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s) + h \cdot N(s))}{\partial N(t)^2} - \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s) - h \cdot N(s))}{\partial N(t)^2} \right) \sigma(t)\, dt \\ &\quad = -2\,\pi\, c(s)\, \sigma(s)\,, \end{aligned} \tag{195}$$

and

$$\begin{aligned} &K^{2,0}_{\gamma,\mathrm{e}}(\sigma)(s) + K^{2,0}_{\gamma,\mathrm{i}}(\sigma)(s) = \\ &\quad = \lim_{h \to 0} \int_0^L \left(\frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s) + h \cdot N(s))}{\partial N(t)^2} + \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s) - h \cdot N(s))}{\partial N(t)^2} \right) \sigma(t)\, dt \\ &\quad = 2 \cdot \text{f.p.} \int_0^L \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s))}{\partial N(t)^2}\, \sigma(t)\, dt\,, \end{aligned} \tag{196}$$

where $c(s)$ denotes the curvature of γ at $\gamma(s)$. In other words, the quadruple layer potential with density σ (see (6)), can be continuously extended from Ω to $\overline{\Omega}$ and from $\mathbb{R}^2 \setminus \overline{\Omega}$ to $\mathbb{R}^2 \setminus \Omega$, with the limiting values given by the formulae

$$p^{2,0}_{\gamma,\sigma,\mathrm{i}}(s) = K^{2,0}_{\gamma,\mathrm{i}}(\sigma)(s) = \pi\, c(s)\, \sigma(s) + \text{f.p.} \int_0^L \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s))}{\partial N(t)^2}\, \sigma(t)\, dt\,, \tag{197}$$

$$p^{2,0}_{\gamma,\sigma,\mathrm{e}}(s) = K^{2,0}_{\gamma,\mathrm{e}}(\sigma)(s) = -\pi\, c(s)\, \sigma(s) + \text{f.p.} \int_0^L \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s))}{\partial N(t)^2}\, \sigma(t)\, dt\,. \tag{198}$$

Proof. Without loss of generality, we can assume that $s \neq 0$ and $s \neq L$. We begin by proving (196). Suppose that $\eta : [0, L] \to \mathbb{R}^2$ is the circle of radius $\frac{L}{2\pi}$ parametrized by its arclength. We define the functions $\Sigma^h_\gamma, \Sigma^h_\eta : [0, L] \times [0, L] \to \mathbb{R}$ via the formulae

$$\Sigma^h_\gamma(s,t) = \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s) + h \cdot N(s))}{\partial N(t)^2} + \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s) - h \cdot N(s))}{\partial N(t)^2}\,, \tag{199}$$

$$\Sigma^h_\eta(s,t) = \frac{\partial^2 \Phi_{\eta(t)}(\eta(s) + h \cdot N(s))}{\partial N(t)^2} + \frac{\partial^2 \Phi_{\eta(t)}(\eta(s) - h \cdot N(s))}{\partial N(t)^2}\,, \tag{200}$$

and, substituting (199), (200) into (196), obtain the identity

$$K^{2,0}_{\gamma,\mathrm{e}}(\sigma)(s) + K^{2,0}_{\gamma,\mathrm{i}}(\sigma)(s) = \lim_{h\to 0} \int_0^L \Sigma^h_\eta(s,t)\, \sigma(t)\, dt + \lim_{h\to 0} \int_0^L \left(\Sigma^h_\gamma(s,t) - \Sigma^h_\eta(s,t)\right) \sigma(t)\, dt\,. \tag{201}$$

Substituting (160), (161) in Theorem 4.3 into (201), we have

$$K^{2,0}_{\gamma,\mathrm{e}}(\sigma)(s) + K^{2,0}_{\gamma,\mathrm{i}}(\sigma)(s) =$$

$$= 2 \cdot \text{f.p.} \int_0^L \frac{\partial^2 \Phi_{\eta(t)}(\eta(s))}{\partial N(t)^2}\, \sigma(t)\, dt + \lim_{h\to 0} \int_0^L \left(\Sigma^h_\gamma(s,t) - \Sigma^h_\eta(s,t)\right) \sigma(t)\, dt\,. \tag{202}$$

Due to Lemma 4.5, there exist positive real constants C_0, a, and h_0 such that for any $s \in [0, L]$

$$\left|\Sigma^h_\gamma(s,t) - \Sigma^h_\eta(s,t)\right| \le C_0\,, \tag{203}$$

for all $|t - s| < a$, $0 \le h < h_0$. For any $t \neq s$ and sufficiently small h, both $\Sigma^h_\gamma(s,t)$ and $\Sigma^h_\eta(s,t)$ are c^∞-functions. Therefore, there also exist positive real constants h_1, C_1 such that for any $s \in [0, L]$

$$\left|\Sigma^h_\gamma(s,t) - \Sigma^h_\eta(s,t)\right| \le C_1\,, \tag{204}$$

for all $|t - s| > a$, $0 \le h < h_1$. Now, applying Lebesgue's dominated convergence theorem (see, for example, [18]) to the second integral of the right hand side of (202), we obtain

$$\lim_{h\to 0} \int_0^L \left(\Sigma^h_\gamma(s,t) - \Sigma^h_\eta(s,t)\right) \sigma(t)\, dt =$$

$$= \int_0^L \lim_{h\to 0} \left(\Sigma^h_\gamma(s,t) - \Sigma^h_\eta(s,t)\right) \sigma(t)\, dt$$

$$= 2 \cdot \int_0^L \left(\frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s))}{\partial N(t)^2} - \frac{\partial^2 \Phi_{\eta(t)}(\eta(s))}{\partial N(t)^2} \right) \sigma(t)\, dt\,. \tag{205}$$

Finally, formula (196) immediately follows from the combination of (202), (205) with (191), (192) in Lemma 4.6.

We now proceed by proving formula (195). We define the functions $\Delta_\gamma^h, \Delta_\eta^h : [0,L]\times[0,L] \to \mathbb{R}$ via the formulae

$$\Delta_\gamma^h(s,t) = \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s)+h\cdot N(s))}{\partial N(t)^2} - \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s)-h\cdot N(s))}{\partial N(t)^2}, \tag{206}$$

$$\Delta_\eta^h(s,t) = \frac{\partial^2 \Phi_{\eta(t)}(\eta(s)+h\cdot N(s))}{\partial N(t)^2} - \frac{\partial^2 \Phi_{\eta(t)}(\eta(s)-h\cdot N(s))}{\partial N(t)^2}, \tag{207}$$

and, by substituting (206), (207) into (195), obtain the identity

$$\begin{aligned} &K_{\gamma,\mathrm{e}}^{2,0}(\sigma)(s) - K_{\gamma,\mathrm{i}}^{2,0}(\sigma)(s) = \\ &= \frac{c(s)\,L}{2\,\pi}\cdot \lim_{h\to 0}\int_0^L \Delta_\eta^h(s,t)\,\sigma(t)\,dt + \lim_{h\to 0}\int_0^L \left(\Delta_\gamma^h(s,t) - \frac{c(s)\,L}{2\,\pi}\cdot\Delta_\eta^h(s,t)\right)\sigma(t)\,dt\,. \end{aligned} \tag{208}$$

Substituting (160), (161) in Theorem 4.3 into (208), we get

$$\begin{aligned} &K_{\gamma,\mathrm{e}}^{2,0}(\sigma)(s) - K_{\gamma,\mathrm{i}}^{2,0}(\sigma)(s) = \\ &\qquad = -2\,\pi\,c(s)\sigma(s) + \lim_{h\to 0}\int_0^L \left(\Delta_\gamma^h(s,t) - \frac{c(s)\,L}{2\,\pi}\cdot\Delta_\eta^h(s,t)\right)\sigma(t)\,dt\,. \end{aligned} \tag{209}$$

Due to Lemma 4.5, there exist positive real constants C_0, a, and h_0 such that for any $s \in [0,L]$

$$\left|\Delta_\gamma^h(s,t) - \frac{c(s)\,L}{2\,\pi}\cdot\Delta_\eta^h(s,t)\right| \le C_0\,, \tag{210}$$

for all $|\,t-s| < a$, $0 \le h < h_0$. For any $t \neq s$ and sufficiently small h, both $\Delta_\gamma^h(s,t)$ and $\Delta_\eta^h(s,t)$ are c^∞-functions. Therefore, there also exist positive real constants h_1, C_1 such that for any $s \in [0,L]$

$$\left|\Delta_\gamma^h(s,t) - \frac{c(s)\,L}{2\,\pi}\cdot\Delta_\eta^h(s,t)\right| \le C_1\,, \tag{211}$$

for all $|\,t-s| > a$, $0 \le h < h_1$. Applying Lebesgue's dominated convergence theorem (see, for example, [18]) to the second integral of the right hand side of (209), we have

$$\lim_{h\to 0}\int_0^L \left(\Delta_\gamma^h(s,t) - \frac{c(s)\,L}{2\,\pi}\cdot\Delta_\eta^h(s,t)\right)\sigma(t)\,dt = \int_0^L \lim_{h\to 0}\left(\Delta_\gamma^h(s,t) - \frac{c(s)\,L}{2\,\pi}\cdot\Delta_\eta^h(s,t)\right)\sigma(t)\,dt\,. \tag{212}$$

Examining (206), (207), we obviously have

$$\lim_{h\to 0}\left(\Delta_\gamma^h(s,t) - \frac{c(s)\,L}{2\,\pi}\cdot\Delta_\eta^h(s,t)\right) = 0\,. \tag{213}$$

Therefore, the integral on the right hand side of (212) is zero, from which (195) follows immediately. □

5 Generalizations

We have presented explicit (modulo an integral operator with a smooth kernel) formulae for integro-pseudodifferential operators of potential theory in two dimensions (up to order 2). The work presented here admits several obvious extensions.

a. Formulae (89) – (107) have their counterparts for elliptic PDEs other than the Laplace equation. Indeed, for any elliptic PDE in two dimensions, the Green's formula has the form

$$G(x,y) = \phi(x,y) \cdot \log(\|x-y\|) + \psi(x,y)\,, \tag{214}$$

with ϕ, ψ a pair of smooth functions; derivations of Section 4 are almost unchanged when $\log(\|x-y\|)$ is replaced with (214). In particular, the counterparts of the formulae (89) – (99) for the Helmholtz equation (with either real or complex Helmholtz coefficient) are identical to (89) – (99); the counterparts of the formulae (100) – (107) for the Helmholtz equation do not coincide with (100) – (107) exactly; instead, they assume the form

$$\begin{aligned}
\text{(a)}\quad K^{3,0}_{\gamma,\mathrm{i}}(\sigma)(s) &= -2\pi\left(c(s)\right)^2\sigma(s) + 4\pi k^2\sigma(s) + \pi\sigma''(s) - 2\pi c'(s)\,H(\sigma)(s) \\
&\quad -3\pi c(s)\,H(\sigma')(s) + N_3(\sigma)(s)\,, && (215)\\
K^{3,0}_{\gamma,\mathrm{e}}(\sigma)(s) &= 2\pi\left(c(s)\right)^2\sigma(s) - 4\pi k^2\sigma(s) - \pi\sigma''(s) - 2\pi c'(s)\,H(\sigma)(s) \\
&\quad -3\pi c(s)\,H(\sigma')(s) + N_3(\sigma)(s)\,, && (216)\\
\text{(b)}\quad K^{2,1}_{\gamma,\mathrm{i}}(\sigma)(s) &= -4\pi k^2\sigma(s) - \pi\sigma''(s) + \pi c'(s)\,H(\sigma)(s) + \pi c(s)\,H(\sigma')(s) \\
&\quad +G_3(\sigma)(s)\,, && (217)\\
K^{2,1}_{\gamma,\mathrm{e}}(\sigma)(s) &= 4\pi k^2\sigma(s) + \pi\sigma''(s) + \pi c'(s)\,H(\sigma)(s) + \pi c(s)\,H(\sigma')(s) \\
&\quad +G_3(\sigma)(s)\,, && (218)\\
\text{(c)}\quad K^{1,2}_{\gamma,\mathrm{i}}(\sigma)(s) &= 4\pi k^2\sigma(s) + \pi\sigma''(s) + \pi c(s)\,H(\sigma')(s) + \widetilde{G_3}(\sigma)(s)\,, && (219)\\
K^{1,2}_{\gamma,\mathrm{e}}(\sigma)(s) &= -4\pi k^2\sigma(s) - \pi\sigma''(s) + \pi c(s)\,H(\sigma')(s) + \widetilde{G_3}(\sigma)(s)\,, && (220)\\
\text{(d)}\quad K^{0,3}_{\gamma,\mathrm{i}}(\sigma)(s) &= 2\pi\left(c(s)\right)^2\sigma(s) - 4\pi k^2\sigma(s) - \pi\sigma''(s) - \pi c'(s)\,H(\sigma)(s) \\
&\quad -3\pi c(s)\,H(\sigma')(s) + \widetilde{N_3}(\sigma)(s)\,, && (221)\\
K^{0,3}_{\gamma,\mathrm{e}}(\sigma)(s) &= -2\pi\left(c(s)\right)^2\sigma(s) + 4\pi k^2\sigma(s) + \pi\sigma''(s) - \pi c'(s)\,H(\sigma)(s) \\
&\quad -3\pi c(s)\,H(\sigma')(s) + \widetilde{N_3}(\sigma)(s)\,, && (222)
\end{aligned}$$

where $k \in \mathbb{C}$ is the Helmholtz coefficient, and the operators N_3, G_3, $\widetilde{N_3}$, $\widetilde{G_3} : L^2[0,L] \to L^2[0,L]$ are compact.

b. The derivation of the three-dimensional counterparts of formulae (89) – (107) is completely straightforward; such expressions have been obtained, and the paper reporting them is in preparation.

c. In certain areas of mathematical physics, one encounters integro-pseudodifferental equations whose analysis is outside the scope of this paper. An important example is the Stratton-Chew equations, to which Maxwell's equations are frequently reduced in computational electromagnetics. Another source of such problems is the scattering of elastic waves in solids. Problems of this type are currently under investigation.

References

[1] A. J. BURTON AND G. F. MILLER, *The Application of Integral Equation Methods to the Numerical Solution of Some Exterior Boundary-value Problems*, Proc. Roy. Soc. Lond. A., 323 (1971), pp. 201–210.

[2] H. CHENG, V. ROKHLIN, AND N. YARVIN, *Non-linear Optimization, Quadrature, and Interpolation*, Tech. Rep. YALEU/DCS/RR-1169, Computer Science Department, Yale University, 1998.

[3] D. COLTON AND R. KRESS, *Integral Equation Methods in Scattering Theory*, John Wiley & Sons, 1983.

[4] M. P. DO CARMO, *Differential Geometry of Curves and Surfaces*, Prentice-Hall, 1976.

[5] M. A. EPTON AND B. DEMBART, *Multipole Translation Theory for the 3-D Laplace and Helmholtz Equations*, SIAM Journal on Scientific Computing, 10 (1995), pp. 865–897.

[6] L. GREENGARD AND V. ROKHLIN, *A New Version of the Fast Multipole Method for the Laplace Equation in Three Dimensions*, Acta Numerica, (1997), pp. 229–269.

[7] J. HADAMARD, *Lectures on the Cauchy's Problem in Linear Partial Differential Equations*, Dover, 1952.

[8] S. KAPUR AND V. ROKHLIN, *High-order Corrected Trapezoidal Rules for Singular Functions*, SIAM Journal of Numerical Analysis, 34 (1997), pp. 1331–1356.

[9] Y. KATZNELSON, *An Introduction to Harmonic Analysis*, Dover, 1976.

[10] P. KOLM AND V. ROKHLIN, *Quadruple and Octuple Layer Potentials in Two Dimensions II: Numerical Techniques*, in preparation.

[11] R. KRESS, *Linear Integral Equations*, Springer, 1989.

[12] J. R. MAUTZ AND R. F. HARRINGTON, *H-field, E-field, and Combined Field Solutions for Conducting Bodies of Revolution*, AEU, 32 (1978), pp. 157–164.

[13] S. G. MIKHLIN, *Integral Equations and Their Applications to Certain Problems in Mechanics, Mathematical Physics and Technology*, Pergamon Press, 1957.

[14] A. F. PETERSON, *The "Interior Resonance" Problem Associated with Surface Integral Equations of Electromagnetics: Numerical Consequences and a Survey of Remedies*, Journal of Electromagnetic Waves and Applications, 10 (1990), pp. 293–312.

[15] V. ROKHLIN, *End-point Corrected Trapezoidal Quadrature Rules for Singular Functions*, Computers and Mathematics with Applications, 20 (1990), pp. 51–62.

[16] A. SIDI AND M. ISRAELI, *Quadrature Methods for Periodic Singular and Weakly Singular Fredholm Integral Equations*, J. Sci. Comp., 3 (1988), pp. 201–231.

[17] J. SONG AND W. C. CHEW, *The Fast Illinois Solver Code: Requirements and Scaling Properties*, IEEE Computational Science and Engineering, (1998), pp. 19–23.

[18] R. L. WHEEDEN AND A. ZYGMUND, *Measure and Integral: An Introduction to Real-Analysis*, Marcel Dekker, 1977.

[19] Y. YAN AND I. H. SLOAN, *On Integral Equations of the First Kind with Logarithmic Kernels*, J. Integral Equations Appl., 1 (1988), pp. 549–579.

[20] S. A. YANG, *Acoustic Scattering by a Hard or Soft Body Across a Wide Frequency Range by the Helmholtz Integral Equation Method*, Journal of the Acoustical Society of America, 102 (1997), pp. 2511–2520.

Numerical Quadratures for Singular and Hypersingular Integrals

P. Kolm and V. Rokhlin
Research Report YALEU/DCS/RR-1190
January 28, 2000

YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

We present a procedure for the design of high order quadrature rules for the numerical evaluation of singular and hypersingular integrals; such integrals are frequently encountered in solution of integral equations of potential theory in two dimensions. Unlike integrals of both smooth and weakly singular functions, hypersingular integrals are pseudo-differential operators, being limits of certain integrals; as a result, standard quadrature formulae fail for hypersingular integrals. On the other hand, such expressions are often encountered in mathematical physics (see, for example, [11]), and it is desirable to have simple and efficient "quadrature" formulae for them. The algorithm we present constructs high-order "quadratures" for the evaluation of hypersingular integrals. The additional advantage of the scheme is the fact that each of the quadratures it produces can be used *simultaneously* for the efficient evaluation of hypersingular integrals, Hilbert transforms, and integrals involving both smooth and logarithmically singular functions; this results in significantly simplified implementations. The performance of the procedure is illustrated with several numerical examples.

Numerical Quadratures for Singular and Hypersingular Integrals

P. Kolm and V. Rokhlin
Research Report YALEU/DCS/RR-1190
January 28, 2000

The first author has been supported in part by DARPA/AFOSR under Contract F49620-97-1-0011. The second author has been supported in part by DARPA/AFOSR under Contract F49620-97-1-0011, in part by ONR under grant N00014-96-1-0188, and in part by AFOSR under Contract F49620-97-C-0052.

Approved for public release: distribution is unlimited.

Keywords: *Numerical Quadrature, Hilbert Transform, Hypersingular Integrals, Pseudo-Differential Operators*

1 Introduction

Numerical integration is one of most frequently encountered computational procedures. When smooth functions are to be integrated, classical techniques tend to be adequate, especially in one and two dimensions; one of most efficient general-purpose tools consists of various versions of nested Gaussian quadrature rules (see, for example, [20, 18, 3, 6]). In cases where extremely efficient special-purpose quadratures are warranted, Gaussian (and more recently, Generalized Gaussian) quadratures are the approach of choice.

When singular functions are to be integrated, the situation tends to be less satisfactory. Special-purpose Gaussian quadratures can be easily constructed for functions of the form

$$f(x) = s(x) \cdot \phi(x), \tag{1}$$

where s is a fixed singular function, and ϕ is smooth. On the other hand, such situations are relatively rare; much more frequently, one is confronted with integrands of the form

$$f(x) = s(x) \cdot \phi(x) + \psi(x) \tag{2}$$

where s is a fixed singular function, and ϕ and ψ are *two* distinct smooth functions (often, several different singularities are involved). Here, Gaussian quadratures can not be used directly, and during the last several years, Generalized Gaussian quadratures have been developed as a tool (in part) for dealing with such situations.

The situation is further complicated when (as frequently happens in potential theory) the "integrals" to be evaluated are not, strictly speaking, integrals, but involve expressions of the form

$$\int_{-1}^{1} \frac{\phi(t)}{y-x}\,dx, \tag{3}$$

$$\int_{-1}^{1} \frac{\phi(x)}{(y-x)^2}\,dx, \tag{4}$$

$$\int_{-1}^{1} \frac{\phi(x)}{(y-x)^3}\,dx, \tag{5}$$

etc., understood in the appropriate finite part sense (in the engineering literature, (4) is often referred to as the "hypersingular" integral). Normally, "integrals" (3) – (5) (and similar objects) are treated via special-purpose techniques (product integration, interpolatory quadratures, etc.). A drawback of this approach is the need to separate singularities of different types, so that each can be treated via an appropriate procedure. For example, in (2), one would need to have access to each of the functions ϕ, ψ individually, as opposed to being able to evaluate the functions *in toto* (the latter situation is frequently encountered in practice).

In this paper, we design a collection of algorithms for the construction of high-order "quadratures" for the evaluation of hypersingular integrals. The additional advantage of the scheme is the fact that each of the quadratures it produces can be used *simultaneously* for the efficient evaluation of hypersingular integrals, Hilbert transforms, and integrals involving both smooth and logarithmically singular functions; this results in significantly simplified implementations.

Remark 1.1 *Unlike the quadratures for functions of the form (2), the quadratures constructed in this paper are not convergent in the classical sense. Instead, they produce a prescribed accuracy for a prescribed set of functions, such as Legendre polynomials, of all orders no greater than some natural number n, Legendre polynomials multiplied by logarithms, etc. Due to the triangle inequality, it is easy to estimate the precision produced when such quadratures are applied to linear combinations of Legendre polynomials, Legendre polynomials multiplied by logarithms, etc. Finally, we observe that if the chosen accuracy is sufficiently small (such as the machine precision), the behavior of the resulting quadratures is indistinguishable from rapid convergence (as can be seen from, for example, Figures 2 - 3 in this paper).*

Remark 1.2 *During the last two decades, numerical techniques have been developed in the computational potential theory (especially, for the Helmholtz equation and related problems involving time-domain Maxwell's equations) that replace classical integral equations with combined integro-pseudo-differential equations. The reasons for these recent developments are involved, and have to do with so-called "spurious resonances" (see, for example, [4, 15, 16, 19]). Without getting into the analytical details, we observe that the interest in the numerical solution of such integro-pseudo-differential equations is growing rapidly, and one of principal motivations behind this work is the design of appropriate rapidly convergent discretization schemes.*

The paper is organized as follows: In Section 2, the necessary mathematical and numerical preliminaries are introduced. In Section 3, we develop numerical quadratures for integrands that are algebraic combinations of smooth functions and functions with singularities of the form $\log|x|$, $\frac{1}{x}$, $\frac{1}{x^2}$. In Section 4, we describe a numerical procedure for the construction of the quadratures from Section 3.2. Section 5 contains numerical examples of some of the quadratures developed in this paper. Finally, in Section 6 we briefly discuss extensions of results of this paper to singularities other than $\log|x|$, $\frac{1}{x}$, $\frac{1}{x^2}$, and to two-dimensional singular and hypersingular integrals.

2 Mathematical and Numerical Preliminaries

In this section, we summarize several results from classical and numerical analysis to be used in the remainder of this paper. Detailed references are given in the text.

2.1 Principal Value Integrals

Integrals of the form

$$\int_a^b \frac{\varphi(x)}{x-y}\, dx\,, \tag{6}$$

where $y \in (a,b)$, do not exist in the classical sense, and are often referred to as *singular integrals.*

Definition 2.1 *Suppose that* φ *is a function* $[a,b] \to \mathbb{R}$, $y \in (a,b)$, *and the limit*

$$\lim_{\epsilon \to 0} \left(\int_a^{y-\epsilon} \frac{\varphi(x)}{x-y}\, dx + \int_{y+\epsilon}^b \frac{\varphi(x)}{x-y}\, dx \right) \tag{7}$$

exists and is finite. Then we will denote the limit (7) by

$$\text{p.v.} \int_a^b \frac{\varphi(x)}{x-y}\, dx\,, \tag{8}$$

and refer to it as a principal value integral.

Theorem 2.1 *Suppose that the function* $\varphi : [a,b] \to \mathbb{R}$ *is continuously differentiable in a neighborhood of* $y \in (a,b)$. *Then the principal value integral (8) exists.*

2.2 Finite Part Integrals

In this paper, we will be dealing with integrals of the form

$$\int_a^b \frac{\varphi(x)}{(x-y)^2}\, dx\,, \tag{9}$$

where $y \in (a,b)$, which are divergent in the classical sense. This type of integrals are often referred to as *hypersingular* or *strongly singular.*

Definition 2.2 *Suppose that* φ *is a function* $[a,b] \to \mathbb{R}$, $y \in (a,b)$, *and the limit*

$$\lim_{\epsilon \to 0} \left(\int_a^{y-\epsilon} \frac{\varphi(x)}{(x-y)^2}\, dx + \int_{y+\epsilon}^b \frac{\varphi(x)}{(x-y)^2}\, dx - \frac{2\varphi(y)}{\epsilon} \right) \tag{10}$$

exists and is finite. Then we will denote the limit (10) by

$$\text{f.p.} \int_a^b \frac{\varphi(x)}{(x-y)^2}\, dx\,, \tag{11}$$

and refer to it as a finite part integral (see, for example, [9]).

The following obvious theorem provides sufficient conditions for the existence of the finite part integral (10), and establishes a connection between finite part and principal value integrals.

Theorem 2.2 *Suppose that the function* $\varphi : [a,b] \to \mathbb{R}$ *is twice continuously differentiable in a neighborhood of* $y \in (a,b)$. *Then the finite part integral (11) exists, and*

$$\text{f.p.} \int_a^b \frac{\varphi(x)}{(x-y)^2}\, dx = \frac{d}{dy}\, \text{p.v.} \int_a^b \frac{\varphi(x)}{x-y}\, dx\,. \tag{12}$$

2.3 Legendre Polynomials and Legendre Expansions

For any natural number n, the Legendre differential equation is

$$(1-x^2)\cdot\frac{d^2u}{dx^2}-2\,x\cdot\frac{du}{dx}+n\,(n+1)\cdot u=0\,. \tag{13}$$

One solution of the Legendre differential equation (13) is the Legendre polynomial $P_n(x):[-1,1]\to\mathbb{R}$, defined by the three-term recursion formula

$$P_{n+1}(x)=\frac{2\,n+1}{n+1}\cdot x\cdot P_n(x)-\frac{n}{n+1}\cdot P_{n-1}(x)\,, \tag{14}$$

with

$$P_0(x) = 1\,, \tag{15}$$

$$P_1(x) = x\,. \tag{16}$$

As is well-known, the Legendre polynomials have an explicit expression given by the formula

$$P_n(x)=\frac{1}{2^n\,n!}\frac{d^n}{dx^n}(x^2-1)^n\,. \tag{17}$$

Furthermore, they are orthogonal with respect to the inner product

$$(f,g)=\int_{-1}^{1}f(x)\,g(x)\,dx\,. \tag{18}$$

Suppose that $x_1,x_2,\dots,x_N$ denote the zeros of the N-th Legendre polynomial $P_N:[-1,1]\to\mathbb{R}$. Then we will refer to the points $\tilde{x}_1,\tilde{x}_2,\dots,\tilde{x}_N$ on the interval $[a,b]$, defined by the formula

$$\tilde{x}_i=\frac{b-a}{2}\cdot x_i+\frac{a+b}{2}\,, \tag{19}$$

for all $i=1,2,\dots,N$, as the N Legendre nodes on $[a,b]$.

For any sufficiently smooth function $\varphi:[-1,1]\to\mathbb{R}$ we will be denoting by a_n the n-th Legendre coefficient of φ, defined by the formula,

$$a_n=\frac{2\,n+1}{2}\int_{-1}^{1}\varphi(x)\,P_n(x)\,dx\,, \tag{20}$$

so that for all $x\in[-1,1]$

$$\varphi(x)=\sum_{n=0}^{\infty}a_n\,P_n(x)\,. \tag{21}$$

The series (21) is referred to as the Legendre expansion of φ. Given any natural number N, for computational purposes we will be approximating the Legendre expansion (21) by its truncated series of degree $N-1$

$$\varphi(x)\approx\sum_{n=0}^{N-1}a_n\,P_n(x)\,. \tag{22}$$

The following lemma states that the truncated Legendre expansion of degree $N-1$ (22) converges rapidly for sufficiently smooth functions, and is proved, for example, in [7].

Lemma 2.3 *Suppose that* $\varphi : [-1,1] \to \mathbb{R}$ *is* k *times continuously differentiable and that* $\sum_{n=0}^{\infty} a_n P_n(x)$ *denotes its Legendre expansion. Then, for any point* $x \in [-1,1]$,

$$\left\| \varphi(x) - \sum_{n=0}^{N-1} a_n P_n(x) \right\|_2 = O\left(\frac{1}{N^k}\right) . \tag{23}$$

The following theorem relates the coefficients in a Legendre expansion to the coefficients in the Legendre expansion of its derivative and integral, respectively. Its proof follows from a combination of results in [21, 1, 7, 8].

Theorem 2.4 *Given a natural number* N, *suppose that the polynomial* $p : [-1,1] \to \mathbb{R}$ *is defined by the formula*

$$p(x) = \sum_{n=0}^{N-1} a_n P_n(x) . \tag{24}$$

Then,

$$p'(x) = \sum_{n=0}^{N-2} b_n P_n(x) , \tag{25}$$

with the coefficients b_n *given by the formula*

$$b_n = (2n+1) \sum_{k=n}^{[\frac{N+n-3}{2}]} a_{2k+1-n} , \quad n = 0, \ldots, N-2 , \tag{26}$$

and with $[\frac{N+n-3}{2}]$ *denoting the integer part of* $\frac{N+n-3}{2}$. *Furthermore,*

$$\int_{-1}^{x} p(y)\, dy = \sum_{n=0}^{N} c_n P_n(x) , \tag{27}$$

with the coefficients c_n *given by the formulae*

$$c_0 = \sum_{n=1}^{N} (-1)^{n+1} c_n , \tag{28}$$

$$c_n = \frac{a_{n-1}}{2(n-1)+1} - \frac{a_{n+1}}{2(n+1)+1} , \quad n = 1, \ldots, N-2 , \tag{29}$$

$$c_{N-1} = \frac{a_{N-2}}{2(N-2)+1} , \tag{30}$$

$$c_N = \frac{a_{N-1}}{2(N-1)+1} . \tag{31}$$

Remark 2.5 *It is well-know that if $\varphi : [-1, 1] \to \mathbb{R}$ is k times continuously differentiable and that $\sum_{n=0}^{\infty} a_n P_n(x)$ denotes its Legendre expansion, then*

$$\left\| \varphi'(x) - \sum_{n=0}^{N-2} b_n P_n(x) \right\|_2 = O\left(\frac{1}{N^{k-1}}\right), \tag{32}$$

and

$$\left\| \int_{-1}^{x} \varphi(y)\, dy - \sum_{n=0}^{N} c_n P_n(x) \right\|_2 = O\left(\frac{1}{N^{k}}\right), \tag{33}$$

where the coefficients b_n and c_n are defined by (26), (28) - (31), respectively.

2.4 Legendre Functions of the Second Kind

The Legendre polynomial P_n (see (17)) is a solution of the Legendre differential equation (13). The other solution is the Legendre function of the second kind $Q_n : \mathbb{C} \setminus [-1, 1] \to \mathbb{C}$, defined by the three-term recursion formula

$$Q_{n+1}(z) = \frac{2n+1}{n+1} \cdot z \cdot Q_n(z) - \frac{n}{n+1} \cdot Q_{n-1}(z), \tag{34}$$

with

$$Q_0(z) = \frac{1}{2} \cdot \log\left(\frac{z+1}{z-1}\right), \tag{35}$$

$$Q_1(z) = \frac{z}{2} \cdot \log\left(\frac{z+1}{z-1}\right) - 1. \tag{36}$$

Clearly, $Q_n(z)$ has a branch cut in the complex z-plane on the real axis from -1 to 1. In agreement with standard practice, on the branch cut we define $Q_n : [-1, 1] \to \mathbb{R}$ by the formula

$$Q_n(x) = \frac{1}{2} \lim_{h \to 0} \Big(Q_n(x + i\,h) + Q_n(x - i\,h) \Big). \tag{37}$$

The following theorem is known as Neumann's integral representation (see, for example, [8]).

Theorem 2.6 *Suppose that $P_n : [-1, 1] \to \mathbb{R}$ denotes the n-th Legendre polynomial, and $Q_n : [-1, 1] \to \mathbb{R}$ the n-th Legendre function of the second kind defined by formula (37). Then, for any point $y \in (-1, 1)$*

$$\text{p.v.} \int_{-1}^{1} \frac{P_n(x)}{y - x}\, dx = 2\, Q_n(y). \tag{38}$$

The following theorem follows immediately from Neumann's integral representation (38) and provides two formulae that will be subsequently used in this paper.

Theorem 2.7 *Suppose that* $P_n : [-1,1] \to \mathbb{R}$ *denotes the* n*-th Legendre polynomial, and* $\widetilde{P}_n : [-1,1] \to \mathbb{R}$ *its primitive function defined by the formula*

$$\widetilde{P}_n(x) = \int_{-1}^{x} P_n(y)\, dy \,. \tag{39}$$

Furthermore, suppose that $Q_n : [-1,1] \to \mathbb{R}$ *denotes the* n*-th Legendre function of the second kind defined by (37). Then, for any point* $y \in (-1,1)$

$$\begin{aligned} \int_{-1}^{1} \frac{1}{2} \cdot \log\left((y-x)^2\right) \cdot P_n(x)\, dx &= \log\left((y-1)^2\right) + \text{p.v.} \int_{-1}^{1} \frac{\widetilde{P}_n(x)}{y-x}\, dx \,, & (40) \\ \text{f.p.} \int_{-1}^{1} \frac{P_n(x)}{(y-x)^2}\, dx &= \text{p.v.} \int_{-1}^{1} \frac{P_n'(x)}{x-y}\, dx + \frac{1}{y-1} - \frac{(-1)^n}{y+1} \,. & (41) \end{aligned}$$

2.5 Chebyshev Systems

Definition 2.3 *A set of continuous functions* $\varphi_1, \ldots, \varphi_N$ *is referred to as a Chebyshev system on the interval* $[a,b]$ *if the determinant*

$$\left| \begin{pmatrix} \varphi_1(x_1) & \cdots & \varphi_1(x_N) \\ \vdots & \ddots & \vdots \\ \varphi_N(x_1) & \cdots & \varphi_N(x_N) \end{pmatrix} \right| \tag{42}$$

is nonzero for any set of points $x_1, \ldots, x_N$ *such that* $a \le x_1 < x_2 < \ldots < x_N \le b$.

Definition 2.4 *Given a set of real numbers* $x_1 \le x_2 \le \ldots \le x_N$, *suppose that* $m_1, m_2, \ldots, m_N$ *denotes the natural numbers defined by the formulae*

$$m_1 = 0 \,, \tag{43}$$

$$m_j = \begin{cases} 0 \,, & \text{for } j > 1 \text{ and } x_j \neq x_{j-1} \,, \\ j-1 \,, & \text{for } j > 1 \text{ and } x_j = x_{j-1} = \ldots = x_1 \,, \\ k \,, & \text{for } j > k+1 \text{ and } x_j = x_{j-1} = \ldots = x_{j-k} \neq x_{j-k-1} \,. \end{cases} \tag{44}$$

A set of continuously differentiable functions $\varphi_1, \ldots, \varphi_N$ *is referred to as an extended Chebyshev system on the interval* $[a,b]$ *if the determinant*

$$\left| \begin{pmatrix} \frac{d^{m_1}}{dx^{m_1}} \varphi_1(x_1) & \cdots & \frac{d^{m_N}}{dx^{m_N}} \varphi_1(x_N) \\ \vdots & \ddots & \vdots \\ \frac{d^{m_1}}{dx^{m_1}} \varphi_N(x_1) & \cdots & \frac{d^{m_N}}{dx^{m_N}} \varphi_N(x_N) \end{pmatrix} \right| , \tag{45}$$

in which $\frac{d^0}{dx^0} \varphi_i(x_j) \equiv \varphi_i(x_j)$, *is nonzero for any set of points* $x_1, \ldots, x_N$ *such that* $a \le x_1 \le x_2 \le \ldots \le x_N \le b$.

Remark 2.8 *Obviously, an extended Chebyshev system also forms a Chebyshev system. The additional constraint is that the points* $x_1, x_2, \ldots, x_N$ *at which the functions are evaluated may be identical. In that case, for each duplicated point, the first corresponding column contains the function values, the second column contains the first derivatives of the functions, the third column contains the second derivatives of the functions, and so forth.*

In the following examples several important cases of Chebyshev and extended Chebyshev systems are presented (additional examples can be found in [10]).

Example 2.1 *The monomials* $1, x, x^2, \ldots, x^n$ *form an extended Chebyshev system on any interval* $[a,b] \subset (-\infty, \infty)$.

Example 2.2 *The exponentials* $e^{-\lambda_1 x}, e^{-\lambda_2 x}, \ldots, e^{-\lambda_n x}$ *form an extended Chebyshev system for any* $\lambda_1, \lambda_2, \ldots, \lambda_n > 0$ *on the interval* $[0, \infty)$.

Example 2.3 *The functions* 1, $\cos(x)$, $\sin(x)$, $\cos(2\,x)$, $\sin(2\,x)$, $\ldots$, $\cos(n\,x)$, $\sin(n\,x)$ *form a Chebyshev system on the interval* $[0, 2\,\pi)$.

2.6 Quadrature Formulae

A quadrature rule on the interval $[-1,1]$ is an expression of the form

$$I_N(\varphi) = \sum_{n=1}^{N} w_n \cdot \varphi(x_n)\,, \tag{46}$$

where the points $x_n \in [-1,1]$ and the coefficients $w_n \in \mathbb{R}$ are referred to as the nodes and the weights of the quadrature, respectively. The quadrature rule $I_N(\varphi)$ serves as an approximation to integrals of the form

$$I(\varphi) = \int_{-1}^{1} w(x) \cdot \varphi(x)\, dx\,, \tag{47}$$

where $\varphi : [-1,1] \to \mathbb{R}$ is a sufficiently smooth function and $w : [-1,1] \to \mathbb{R}$ is some fixed weight function. Since we will permit the function w to be strongly singular, the integral (47) has to be evaluated in the appropriate sense. In particular, for $w(x)$ we will consider, *inter alia*, the singular functions

$$\frac{1}{2} \cdot \log\left((y-x)^2\right), \tag{48}$$

$$\frac{1}{y-x}\,, \tag{49}$$

$$\frac{1}{(y-x)^2}\,, \tag{50}$$

where $y \in (-1,1)$. For the latter two functions, the integral (47) is interpreted as a principal value integral (see (7)) and finite part integral (see (10)), respectively.

Definition 2.5 *A quadrature formula (46) for the integral (47) is said to be of the degree* $M \geq 1$, *if it integrates all polynomials up to degree* M *exactly.*

Normally, the degree of a quadrature formula (46) can not exceed $2N-1$ (see, for example, [20]). Quadrature rules (46) of degree $2N-1$ are commonly referred to as *Gaussian quadrature rules.* The following theorem is well-known and can be found in most elementary textbooks on numerical analysis (see, for example, [20]).

Theorem 2.9 *(Gaussian quadrature) Suppose that* $w(x) \equiv 1$ *for all* $x \in [-1,1]$. *Then there exists a unique quadrature rule (46) which has the degree* $2N-1$. *Furthermore, the nodes* $x_1, x_2, \ldots, x_N$ *are the zeros of the* N*-th Legendre polynomial* $P_N(x)$ *(see, (17)), and the weights* $w_1, w_2, \ldots, w_N$ *are all positive and given by the formula*

$$w_n = \int_{-1}^{1} \prod_{\substack{j=1 \\ j \neq n}}^{N} \left(\frac{x - x_j}{x_n - x_j} \right)^2 dx , \quad n = 1, 2, \ldots, N . \tag{51}$$

2.7 Generalized Gaussian Quadrature

Numerical quadratures are normally constructed such that the quadrature rule (46) is *exactly* equal to the integral (47) for some set of functions. Classical N-point Gaussian quadratures (see, Theorem 2.9) integrate polynomials of order $2N-1$ exactly. In [14], the notion of Gaussian quadrature was generalized as follows.

Definition 2.6 *Suppose that* $w : [-1,1] \to \mathbb{R}$ *is a non-negative integrable function. A quadrature rule (46) will be referred to as Gaussian with the respect to a set of* $2N$ *functions* $\varphi_1, \varphi_2, \ldots, \varphi_{2N} : [-1,1] \to \mathbb{R}$ *and a weight function* w, *if it consists of* N *weights and nodes, and integrates the functions* $w \circ \varphi_i$ *on* $[-1,1]$ *exactly for all* $i = 1, 2, \ldots, 2N$. *The weights and the nodes of a Gaussian quadrature will be referred to as Gaussian weights and nodes, respectively.*

The following theorem states that the Gaussian quadrature with respect to a set of functions $\varphi_1, \varphi_2, \ldots, \varphi_{2N}$ exists and is unique if the set $\varphi_1, \varphi_2, \ldots, \varphi_{2N}$ forms a Chebyshev system (see Definition 2.3). It is proved (in a slightly different form) in [10, 13].

Theorem 2.10 *Suppose that the functions* $\varphi_1, \varphi_2, \ldots, \varphi_{2N} : [-1,1] \to \mathbb{R}$ *form a Chebyshev system (see Definition 2.3) on the interval* $[-1,1]$, *and that the weight function* $w : [-1,1] \to \mathbb{R}$ *is non-negative and integrable. Then there exists a unique Gaussian quadrature with respect to the set* $\varphi_1, \varphi_2, \ldots, \varphi_{2N}$ *and the weight function* w. *Furthermore, the weights of this quadrature are all positive.*

From Definition 2.6 it immediately follows that the Gaussian quadrature with respect to the functions $\varphi_1, \varphi_2, \ldots, \varphi_{2N} : [-1,1] \to \mathbb{R}$ and the weight function $w : [-1,1] \to \mathbb{R}$ is defined by the system of equations

$$\begin{aligned}
\sum_{n=1}^{N} w_n \cdot \varphi_1(x_n) &= \int_{-1}^{1} w(x) \cdot \varphi_1(x)\, dx\,, \\
\sum_{n=1}^{N} w_n \cdot \varphi_2(x_n) &= \int_{-1}^{1} w(x) \cdot \varphi_2(x)\, dx\,, \\
&\vdots \\
\sum_{n=1}^{N} w_n \cdot \varphi_{2N}(x_n) &= \int_{-1}^{1} w(x) \cdot \varphi_{2N}(x)\, dx\,. \qquad (52)
\end{aligned}$$

We denote the left hand sides of these equations by $f_1, f_2, \ldots, f_{2N}$; each of the f_i's being a function $[-1,1]^N \times \mathbb{R}^N \to \mathbb{R}$ of the nodes $x_1, x_2, \ldots, x_N$ and weights $w_1, w_2, \ldots, w_N$, respectively. Their partial derivatives are given by the formulae

$$\frac{\partial f_i}{\partial w_n} = \varphi_i(x_n)\,, \qquad (53)$$

$$\frac{\partial f_i}{\partial x_n} = w_n\, \varphi_i'(x_n)\,, \qquad (54)$$

so that the Jacobian of the system (52) takes the form

$$J(x_1, \ldots, x_N, w_1, \ldots, w_N) =$$
$$\begin{pmatrix} \varphi_1(x_1) & \cdots & \varphi_1(x_N) & w_1\, \varphi_1'(x_1) & \cdots & w_N\, \varphi_1'(x_N) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \varphi_{2N}(x_1) & \cdots & \varphi_{2N}(x_N) & w_1\, \varphi_{2N}'(x_1) & \cdots & w_N\, \varphi_{2N}'(x_N) \end{pmatrix}. \qquad (55)$$

In practice, the system (52) is solved via Newton's method (see, for example, [5]). The following theorem states that when the functions to be integrated constitute an extended Chebyshev system, Newton's method for this system is always quadratically convergent, provided the starting point for the iteration is within a sufficiently small neighborhood of the solution. A proof can be found in, for example, [5].

Theorem 2.11 *Suppose that the functions* $\varphi_1, \varphi_2, \ldots, \varphi_{2N}$ *form an extended Chebyshev system (see Definition 2.4). Suppose further that the Gaussian quadrature nodes and weights for these functions are denoted by* $\widehat{x}_1, \widehat{x}_2, \ldots, \widehat{x}_N$ *and* $\widehat{w}_1, \widehat{w}_2, \ldots, \widehat{w}_N$*, respectively. Then the determinant of the Jacobian matrix (55) is nonzero at the point* $(\widehat{x}_1, \ldots, \widehat{x}_N, \widehat{w}_1, \ldots, \widehat{w}_N)$*, i.e.*

$$|J(\widehat{x}_1, \ldots, \widehat{x}_N, \widehat{w}_1, \ldots, \widehat{w}_N)| \neq 0\,. \qquad (56)$$

Furthermore, the nodes $\widehat{x}_1, \widehat{x}_2, \ldots, \widehat{x}_N$ *and the weights* $\widehat{w}_1, \widehat{w}_2, \ldots, \widehat{w}_N$ *depend continuously on the weight function* w.

Remark 2.12 *In order for Newton's method to converge, the starting point must be within a sufficiently small neighborhood of the solution. In [5] the continuation method (sometimes also referred to as the homotopy method) is used to generate such starting points.*

2.8 Singular Value Decomposition of a Set of Functions

The following theorem generalizes the standard singular value decomposition of a matrix to a set of functions. A proof can be found (in a more general form), for example, in [17].

Theorem 2.13 *Suppose that the functions* $\varphi_1, \varphi_2, \ldots, \varphi_N : [a,b] \to \mathbb{R}$ *are square integrable. Then for some integer* M *there exist an orthonormal set of functions* $u_1, u_2, \ldots, u_M : [a,b] \to \mathbb{R}$, *an* $N \times M$ *matrix* $V = [v_{ij}]$ *with orthonormal columns, and a set of real numbers* $s_1 \geq s_2 \geq \ldots \geq s_M > 0$, *such that*

$$\varphi_j(x) = \sum_{i=1}^{M} u_i(x)\, s_i\, v_{ij}\,, \tag{57}$$

for all $x \in [a,b]$ *and all* $n = 1, 2, \ldots, N$.

By analogy to the well-known singular value decomposition of matrices, we will refer to the factorization (57) as the singular value decomposition of the set of functions $\varphi_1, \varphi_2, \ldots, \varphi_N$, the functions $u_1, u_2, \ldots, u_M$ as the singular functions, the columns of the matrix V as singular vectors, and the numbers $s_1 \geq s_2 \geq \ldots \geq s_M$ as the singular values, respectively.

The following theorem from [5] states that the accuracy of a quadrature formula with positive weights for the functions $\varphi_1, \varphi_2, \ldots, \varphi_N$ is determined by its accuracy for the singular functions u_i, corresponding to non-trivial singular values.

Theorem 2.14 *Suppose that under the conditions of Theorem 2.13 there exist a positive real number* ϵ *and an integer* $1 < M_0 < M$, *such that*

$$\sum_{i=M_0+1}^{M} s_i^2 < \frac{\epsilon^2}{4}\,. \tag{58}$$

Suppose further that the L-point quadrature rule with nodes $x_1, x_2, \ldots, x_L$ *and weights* $w_1, w_2, \ldots, w_L$ *integrates the functions* u_i *exactly on the interval* $[a,b]$, *i.e.*

$$\sum_{j=1}^{L} w_j \cdot u_i(x_j) = \int_a^b u_i(x)\, dx \tag{59}$$

for all $i = 1, 2, \ldots, M_0$, *and that the weights* $w_1, w_2, \ldots, w_L$ *are all positive. Then for each* $i = 1, 2, \ldots, N$,

$$\left| \sum_{j=1}^{L} w_j \cdot \varphi_i(x_j) - \int_a^b \varphi_i(x)\, dx \right| < \epsilon \cdot \|\varphi_i\|_2\,. \tag{60}$$

3 Analytical Apparatus

The principal purpose of this paper is to construct quadrature formulae for functions $f : [-1,1] \to \mathbb{R}$ of the form

$$f(x) = \varphi(x) + \psi(x) \cdot \log|x| + \frac{\eta(x)}{x} + \frac{\theta(x)}{x^2}\,, \tag{61}$$

where $\varphi, \psi, \eta, \theta : [-1,1] \to \mathbb{R}$ are smooth. In Section 3.1, we construct separate quadrature formulae for each of the functions of the form

$$\varphi(x)\,, \quad \psi(x)\cdot\log|x|\,, \quad \frac{\eta(x)}{x}\,, \quad \frac{\theta(x)}{x^2}\,, \tag{62}$$

in Section 3.2, we present a scheme were each quadrature it produces can be used *simultaneously* for the efficient numerical integration of functions of the form (61).

Obviously, integrals of the form

$$\int_{-1}^{1}\Big(\varphi(x) + \psi(x)\cdot\log(|y-x|) + \frac{\eta(x)}{y-x} + \frac{\theta(x)}{(y-x)^2}\Big)\,dx \tag{63}$$

with y *outside* the interval of integration $[-1,1]$ and the functions $\varphi, \psi, \eta, \theta$ smooth, can be evaluated with standard Gaussian quadrature formulae. However, when y is sufficiently close to the interval of integration $[-1,1]$, the number of Gaussian nodes needed to achieve acceptable accuracy is often very high. Therefore, more specialized quadratures are desirable in this case; Section 3.3 is devoted to the design of generalized Gaussian quadratures for this environment.

3.1 Quadrature Formulae for Individual Singularities $\log|x|$, $\frac{1}{x}$, $\frac{1}{x^2}$

The following theorem is one of principal analytical tools used in this paper.

Theorem 3.1 *Suppose that* $x_1, x_2, \ldots, x_N$ *and* $w_1, w_2, \ldots, w_N$ *denote the* N *nodes and weights of the Gaussian quadrature on the interval* $[-1,1]$*, respectively (see, Theorem 2.9). Suppose further that* $P_j(x)$ *denotes the* j*-th Legendre polynomial (see, (17)), and that* $w(x)\cdot P_j(x)$ *is integrable on* $[-1,1]$ *for all* $j = 0, 1, \ldots, N-1$*. Then the quadrature rule*

$$\int_{-1}^{1} w(x)\cdot\varphi(x)\,dx \approx \sum_{n=1}^{N}\tilde{w}_n\cdot\varphi(x_n) \tag{64}$$

with the weights $\tilde{w}_n$ *defined by the formula*

$$\tilde{w}_n = w_n\cdot\sum_{j=0}^{N-1}\Big(\frac{2j+1}{2}\,P_j(x_n)\cdot\Big(\int_{-1}^{1} w(x)\,P_j(x)\,dx\Big)\Big) \tag{65}$$

has the degree $N-1$.

Proof. Suppose that $\varphi : [-1,1] \to \mathbb{R}$ is a polynomial of order $N-1$ given by its Legendre series (21) so that

$$\varphi(x) = \sum_{j=0}^{N-1} a_j\,P_j(x)\,. \tag{66}$$

Substituting (66) into (47), we obtain

$$I(\varphi) = \int_{-1}^{1} w(x) \cdot \varphi(x)\, dx = \int_{-1}^{1} w(x) \cdot \Big(\sum_{j=0}^{N-1} a_j\, P_j(x) \Big)\, dx = \sum_{j=0}^{N-1} a_j \cdot \Big(\int_{-1}^{1} w(x)\, P_j(x)\, dx \Big) . \tag{67}$$

The coefficients a_j are given by (20). Evaluating the integral (20) via N-point Gaussian quadrature (see Theorem 2.9), we obtain the identity

$$a_j = \frac{2j+1}{2} \int_{-1}^{1} \varphi(x)\, P_j(x)\, dx = \frac{2j+1}{2} \sum_{n=1}^{N} w_n \cdot \varphi(x_n) \cdot P_j(x_n) , \tag{68}$$

for all $j = 0, 1, \ldots, N-1$. Finally, substituting (68) into (67) we obtain

$$\int_{-1}^{1} w(x) \cdot \varphi(x)\, dx = \sum_{n=1}^{N} \varphi(x_n) \cdot w_n \cdot \sum_{j=0}^{N-1} \Big(\frac{2j+1}{2} P_j(x_n) \cdot \Big(\int_{-1}^{1} w(x)\, P_j(x)\, dx \Big) \Big) , \tag{69}$$

from which (64) and (65) immediately follow. □

Remark 3.2 *If the function φ is k times continuously differentiable, it immediately follows from the Cauchy-Schwartz inequality and (23) in Lemma 2.3 that*

$$\Big| \int_{-1}^{1} w(x) \cdot \varphi(x)\, dx - \sum_{n=1}^{N} \tilde{w}_n \cdot \varphi(x_n) \Big| = O\Big(\frac{1}{N^k} \Big) . \tag{70}$$

The following theorem extends Theorem 3.1 to the case when the function $w : [-1,1] \to \mathbb{R}$ is defined by one of the formulae (48) – (50). The latter two functions are not integrable in the classical sense, and the integral (47) is interpreted as a principal value integral (see (7)) and finite part integral (see (10)), respectively. The theorem follows immediately from the combination of Theorems 2.4, 2.7, 3.1.

Theorem 3.3 *Suppose that $x_1, x_2, \ldots, x_N$ and $w_1, w_2, \ldots, w_N$ denote the N nodes and weights of the Gaussian quadrature on the interval $[-1,1]$ (see, Theorem 2.9). Suppose further that $\varphi : [-1,1] \to \mathbb{R}$ is a sufficiently smooth function, and $P_j(x)$, $Q_j(x)$ denote the j-th Legendre polynomial and Legendre function of the second kind (see, (17), (37)), respectively. Finally, suppose that the coefficients $w_{1,1}, w_{1,2}, \ldots, w_{1,N}$, $w_{2,1}, w_{2,2}, \ldots, w_{2,N}$, $w_{3,1}, w_{3,2}, \ldots, w_{3,N}$, are defined by the formulae*

$$w_{1,n} = w_n \cdot \sum_{j=0}^{N-1} (2j+1) \cdot P_j(x_n) \cdot Q_j(y) , \tag{71}$$

$$w_{2,n} = w_n \cdot \Big(\Big(P_0(x_n) - P_1(x_n) \Big) \cdot R_0(y) + \sum_{j=1}^{N-2} \Big(P_{j-1}(x_n) - P_{j+1}(x_n) \Big) \cdot R_j(y)$$

$$+P_{N-2}(x_n)\cdot R_{N-1}(y)+P_{N-1}(x_n)\cdot R_N(y)\Big), \tag{72}$$

$$\begin{aligned} w_{3,n} &= w_n\cdot\Bigg(-\sum_{j=0}^{N-2}\sum_{k=j}^{[\frac{N+j-3}{2}]}(2\,j+1)\cdot(4\,k+3-2\,n)\cdot Q_j(y)\cdot P_{2\,k+1-n}(x_n) \\ &\quad +\sum_{j=0}^{N-1}\frac{2\,j+1}{2}\,P_j(x_n)\cdot\left(\frac{1}{y-1}-\frac{(-1)^j}{y+1}\right)\Bigg); \end{aligned} \tag{73}$$

for all $n=1,2,\ldots,N$, *with* $[\frac{N+j-3}{2}]$ *denoting the integer part of* $\frac{N+j-3}{2}$, *and the mappings* $R_j:(-1,1)\to\mathbb{R}$ *defined by the formula*

$$R_j(y)=Q_j(y)+\frac{1}{4}\cdot\log\left((y-1)^2\right). \tag{74}$$

Then, for any point $y\in(-1,1)$, *the quadrature rules*

$$\text{p.v.}\int_{-1}^{1}\frac{\varphi(x)}{y-x}\,dx\approx\sum_{n=1}^{N}w_{1,n}\cdot\varphi(x_n), \tag{75}$$

$$\int_{-1}^{1}\frac{1}{2}\cdot\log\left((y-x)^2\right)\cdot\varphi(x)\,dx\approx\sum_{n=1}^{N}w_{2,n}\cdot\varphi(x_n), \tag{76}$$

$$\text{f.p.}\int_{-1}^{1}\frac{\varphi(x)}{(y-x)^2}\,dx\approx\sum_{n=1}^{N}w_{3,n}\cdot\varphi(x_n), \tag{77}$$

have the degree $N-1$, $N-2$, *and* $N-1$, *respectively.*

3.2 Quadrature Formulae for Functions of the Form $\varphi(x)+\psi(x)\cdot\log|x| +\frac{\eta(x)}{x}+\frac{\theta(x)}{x^2}$

Theorem 3.3 provides a tool for the numerical integration of functions of the form

$$\psi(x)\cdot\log|x|, \tag{78}$$

$$\frac{\eta(x)}{x}, \tag{79}$$

$$\frac{\theta(x)}{x^2}. \tag{80}$$

However, integrands are frequently encountered of the form

$$f(x)=\varphi(x)+\psi(x)\cdot\log|x|+\frac{\eta(x)}{x}+\frac{\theta(x)}{x^2}, \tag{81}$$

where the functions φ,ψ,η,θ are known to be smooth but are not available individually. Specifically, in the numerical solution of scattering problems, one is frequently confronted with the need to evaluate integrals of the form

$$\begin{aligned} \text{f.p.}\int_{-1}^{1} K(x,y)\cdot\sigma(x)\,dx = \\ = \text{f.p.}\int_{-1}^{1}\left(K_1(x,y)+K_2(x,y)\cdot\log(|x-y|)+\frac{K_3(x,y)}{y-x}+\frac{K_4(x,y)}{(y-x)^2}\right)\cdot\sigma(x)\,dx\,, \end{aligned} \qquad (82)$$

where $\sigma : [-1,1] \to \mathbb{R}$ and $K_1(x,y), K_2(x,y), K_3(x,y), K_4(x,y) : [-1,1]\times[-1,1] \to \mathbb{R}$ are smooth functions, and $y \in (-1,1)$. Normally, the functions K_1, K_2, K_3, K_4 are not available separately, so that only the kernel K *in toto* can be evaluated. In such cases, a single quadrature rule integrating functions of the composite form (81) is clearly preferable. Even when each of the functions φ, ψ, η, θ is available separately, the numerical implementation is simplified when a single quadrature formula can be used.

Given a real number $y \in (-1,1)$, we denote by $\psi_1, \psi_2, \ldots, \psi_{4M}$ the functions $[-1,1] \to \mathbb{R}$ defined by the formulae

$$\psi_i(x) = \begin{cases} P_{i-1}(x)\,, & \text{for } i = 1,\ldots,M\,, \\ P_{i-M-1}(x)\cdot\log(|y-x|)\,, & \text{for } i = M+1,\ldots,2M\,, \\ P_{i-2M-1}(x)\cdot\dfrac{1}{y-x}\,, & \text{for } i = 2M+1,\ldots,3M\,, \\ P_{i-3M-1}(x)\cdot\dfrac{1}{(y-x)^2}\,, & \text{for } i = 3M+1,\ldots,4M\,. \end{cases} \qquad (83)$$

In a minor generalization of the standard terminology, we define the generalized moments $m_1(y)$, $m_2(y), \ldots, m_{4M}(y)$ by the formulae

$$m_i(y) = \begin{cases} \displaystyle\int_{-1}^{1} P_{i-1}(x)\,dx\,, & \text{for } i = 1,\ldots,M\,, \\ \displaystyle\int_{-1}^{1} P_{i-M-1}(x)\cdot\log(|y-x|)\,dx\,, & \text{for } i = M+1,\ldots,2M\,, \\ \text{p.v.}\displaystyle\int_{-1}^{1} \frac{P_{i-2M-1}(x)}{y-x}\,dx\,, & \text{for } i = 2M+1,\ldots,3M\,, \\ \text{f.p.}\displaystyle\int_{-1}^{1} \frac{P_{i-3M-1}(x)}{(y-x)^2}\,dx\,, & \text{for } i = 3M+1,\ldots,4M\,. \end{cases} \qquad (84)$$

Now, suppose that $x_1, x_2, \ldots, x_N$ denotes the N Legendre nodes on $[-1,1]$ (see (19)). Then we define the weights $w_1, w_2, \ldots, w_N$ of the quadrature formula

$$\int_{-1}^{1} f(x)\,dx \approx \sum_{n=1}^{N} w_n\cdot f(x_n) \qquad (85)$$

as the solution of the system of the $4M$ linear algebraic equations

$$\sum_{n=1}^{N} w_n\cdot\psi_1(x_n) = m_1(y)\,,$$

$$\begin{aligned}\sum_{n=1}^{N} w_n \cdot \psi_2(x_n) &= m_2(y)\,, \\ &\vdots \\ \sum_{n=1}^{N} w_n \cdot \psi_{4M}(x_n) &= m_{4M}(y)\,.\end{aligned} \tag{86}$$

Obviously, the matrix of the system (86) might be square, or it might be over- or underdetermined, depending on the values of the parameters M, N. On the other hand, given a solution $w_1, w_2, \ldots, w_n$ of (86), we can be sure that the quadrature formula (85) will integrate exactly all functions f of the form (81), as long as the functions φ, ψ, η, θ are polynomials of order not greater than $M-1$. Due to Theorem A.6 in Appendix A below, for sufficiently large N, there always exist multiple solutions of (86), and a solution $\tilde{w}_1, \tilde{w}_2, \ldots, \tilde{w}_n$ can be found such that

$$\sum_{n=1}^{N} \tilde{w}_n^2 \leq C \cdot \sum_{n=1}^{N} w_n^2\,, \tag{87}$$

where $w_1, w_2, \ldots, w_N$ are the weights of the N-point Gaussian quadrature and C is a positive real constant. In practice, least squares are used to find $w_1, w_2, \ldots, w_n$ satisfying the bound (87) (see Section 4 below). Denoting the $N \times 4M$ matrix of system (86) by A and its right-hand side by b, we rewrite (86) in the form

$$A\,w = b\,. \tag{88}$$

3.3 Generalized Gaussian Quadrature Formulae for Functions of the Form $\varphi(x) + \psi(x) \cdot \log|x| + \dfrac{\eta(x)}{x} + \dfrac{\theta(x)}{x^2}$

In Section 3.2 we described the quadrature formula (85) for integrals of the form (82) where the point of evaluation y is *inside* the interval of integration. While standard numerical quadratures (eg. Newton-Cotes or Gaussian quadratures) can be used for integrals of the form (82) when the point of evaluation y is *outside and sufficiently far away* from the interval of integration, more specialized quadratures are desirable when y is *outside but close* to the interval of integration.

Given two positive real numbers d and R such that $d < R$, we will denote by $D_{R,d}$ the set $[-R, -1-d] \cup [1+d, R]$ (see Figure 1). We define the functions ψ_1, ψ_2, ..., $\psi_{4M} : [-1,1] \times D_{R,d} \to \mathbb{R}$ by the formulae

$$\psi_i(x,y) = \begin{cases} P_{i-1}(x)\,, & \text{for } i = 1, \ldots, M\,, \\ P_{i-M-1}(x) \cdot \log(|y-x|)\,, & \text{for } i = M+1, \ldots, 2M\,, \\ P_{i-2M-1}(x) \cdot \dfrac{1}{y-x}\,, & \text{for } i = 2M+1, \ldots, 3M\,, \\ P_{i-3M-1}(x) \cdot \dfrac{1}{(y-x)^2}\,, & \text{for } i = 3M+1, \ldots, 4M\,, \end{cases} \tag{89}$$

where P_j denotes the j-th Legendre polynomial (17).

Now, suppose that $y_1, y_2, \ldots, y_K$ are points in $D_{R,d}$. We will denote by $\eta_{ij} : [-1,1] \to \mathbb{R}$ the $4 \cdot K \cdot M$ functions defined by the formula

$$\eta_{ij}(x) = \psi_i(x, y_j) \tag{90}$$

where $i = 1, 2, \ldots, 4M$ and $j = 1, 2, \ldots, K$. Since it will be convenient to view the functions η_{ij} as a finite sequence of functions $[-1,1] \to \mathbb{R}$, we introduce the notation

$$k = 4\,(j-1)\,M + i\,, \tag{91}$$

so that

$$i = k - 4\,(j-1)\,M\,, \tag{92}$$

$$j = \frac{k-i}{4\,M} + 1\,. \tag{93}$$

In a mild abuse of notation, we will use η_k and η_{ij} interchangeably.

Due to Theorem 2.13, there exist orthonormal functions $u_1, u_2, \ldots, u_L : [-1,1] \to \mathbb{R}$, a matrix $V \in \mathbb{R}^{4 \cdot K \cdot M \times L}$ with orthonormal columns, and real numbers $s_1 \geq s_2 \geq \cdots \geq s_L > 0$, for some integer $L \leq 4 \cdot K \cdot M$, such that

$$\eta_k(x) = \sum_{i=1}^{L} u_i(x)\, s_i\, v_{ik} \tag{94}$$

for all $k = 1, 2, \ldots, 4 \cdot K \cdot M$.

Remark 3.4 *For an arbitrary positive real number ϵ, we will denote by $n(\epsilon)$ the number of coefficients s_i in the decomposition (94) such that $s_i > \epsilon$. It turns out that for fixed d and R, $n(\epsilon)$ is proportional to $\log(\frac{1}{\epsilon})$, and is virtually independent of K. For a fixed ϵ, $n(\epsilon)$ is proportional to $\log(\frac{R}{d})$, and is virtually independent of K. The behavior of $n(\epsilon)$ as a function ϵ, d, R is investigated in detail in [22].*

The following theorem is an immediate consequence of Theorems 2.11, 2.14.

Theorem 3.5 *Suppose that for a sufficiently large integer number K, $y_1, y_2, \ldots, y_K$ are points in $D_{R,d}$ such that $y_i \neq y_j$ for all $i \neq j$. Suppose further that the functions η_1, η_2, ..., $\eta_{4KM} : [-1,1] \to \mathbb{R}$, the real positive numbers $s_1, s_2, \ldots, s_L$, and the functions $u_1, u_2, \ldots, u_L : [-1,1] \to \mathbb{R}$ are defined by the formulae (90), (94), respectively. Given a positive real number ϵ, we denote by L_0 the smallest even integer such that $1 < L_0 < L$ and*

$$\sum_{i=L_0+1}^{L} s_i^2 < \frac{\epsilon^2}{4}\,. \tag{95}$$

Then there exists a unique solution $(w_1, \dots w_{\frac{L_0}{2}}, x_1, \dots x_{\frac{L_0}{2}})$ *of the non-linear system*

$$\begin{aligned}
\sum_{n=1}^{\frac{L_0}{2}} w_n \cdot u_1(x_n) &= \int_{-1}^{1} u_1(x)\,dx\,, \\
\sum_{n=1}^{\frac{L_0}{2}} w_n \cdot u_2(x_n) &= \int_{-1}^{1} u_2(x)\,dx\,, \\
&\vdots \\
\sum_{n=1}^{\frac{L_0}{2}} w_n \cdot u_{L_0}(x_n) &= \int_{-1}^{1} u_{L_0}(x)\,dx\,,
\end{aligned} \tag{96}$$

where all w_n, $n = 1, 2, \dots, \frac{L_0}{2}$, *are positive. Furthermore, for each* $k = 1, 2, \dots, 4 \cdot K \cdot M$, *the* $\frac{L_0}{2}$*-point quadrature rule*

$$\sum_{n=1}^{\frac{L_0}{2}} w_n \cdot \eta_k(x_n) \approx \int_{-1}^{1} \eta_k(x)\,dx\,, \tag{97}$$

has relative accuracy ϵ*; that is*

$$\left| \sum_{n=1}^{\frac{L_0}{2}} w_n \cdot \eta_k(x_n) - \int_{-1}^{1} \eta_k(x)\,dx \right| < \epsilon \cdot \|\eta_k\|_2\,. \tag{98}$$

Remark 3.6 *The solution of the system of non-linear equations (96) can be found by Newton's method. For a detailed discussion of a Newton method for non-linear systems arising in the construction of generalized Gaussian quadratures, the reader is referred to [5].*

4 Numerical Algorithm

In Sections 3.1, 3.2 we have described quadratures rules for integrands of the form (78) – (81). While the numerical evaluation of the weights of the quadratures (75) – (77) in Section 3.1 via the formulae (72) – (73) is straightforward, the evaluation of the weights $w_1, w_2, \dots, w_N$ of the quadrature (85) is more involved; we summarize the computational procedure below.

The input to the algorithm is a real number $y \in (-1, 1)$, a natural number N where N is the number of Legendre nodes (19) on the interval $[-1, 1]$, and a natural number M where $M - 1$ is the degree of the quadrature rule. The algorithm will then compute quadrature weights $w_1, w_2, \dots, w_N$, such that

$$\sum_{n=1}^{N} w_n \cdot \varphi(x_n) \approx \int_{-1}^{1} w(x) \cdot \varphi(x)\,dx\,, \tag{99}$$

where $\varphi : [-1,1] \to \mathbb{R}$ is smooth and $w : [-1,1] \to \mathbb{R}$ is a linear combination of smooth functions and functions of the form (48) – (50), respectively. It consists of the following steps:

1. Construct the N-point Gaussian nodes $x_1, x_2, \ldots, x_N$ and weights $w_1, w_2, \ldots w_N$ on the interval $[-1,1]$ (see Theorem 2.9).
2. Evaluate the Legendre polynomials $P_0, P_1, \ldots, P_{M-1}$ at the nodes $x_1, x_2, \ldots, x_N$ via the three-term recursion (14).
3. Evaluate all the functions $\psi_1, \psi_2, \ldots, \psi_{4M}$ (see (83)) at the nodes $x_1, x_2, \ldots, x_N$.
4. Construct the moments $m_1(y), m_2(y), \ldots, m_{4M}(y)$ (see (84)) exactly, using Gaussian quadrature for m_1, m_2, ..., m_M and quadrature rules (75) – (77) for $m_{M+1}(y)$, $m_{M+2}(y)$, ..., $m_{4M}(y)$, respectively.
5. Solve the linear algebraic system (88) in the least squares sense with any standard routine (available, for example, in LAPACK [2]).

5 Numerical Examples

FORTRAN codes have been written constructing the quadratures described in Sections 3.1, 3.2, 3.3; in this section, their performance is illustrated with several numerical examples. In all examples below the quadrature nodes and weights are first computed in extended precision arithmetic (`REAL *16`) to assure full double precision accuracy. The quadrature rules are then used in double precision (`REAL *8`) to numerically integrate a number of functions with singularities $\log|x|$, $\frac{1}{x}$, $\frac{1}{x^2}$.

Example 5.1 In the first example, we use the quadrature rules (75) – (77) to evaluate integrals of the form (47) for each of the singularities (48) – (50) with the function $\varphi : [-1,1] \to \mathbb{R}$ defined by the formula

$$\varphi(x) = \sin(2\,x) + \cos(3\,x)\,, \tag{100}$$

so that the actual functions to be integrated are of the form

$$\log(|x-y|) \cdot \Big(\sin(2\,x) + \cos(3\,x)\Big)\,, \tag{101}$$

$$\frac{1}{y-x} \cdot \Big(\sin(2\,x) + \cos(3\,x)\Big)\,, \tag{102}$$

$$\frac{1}{(y-x)^2} \cdot \Big(\sin(2\,x) + \cos(3\,x)\Big)\,. \tag{103}$$

We denote by $y_1, y_2, \ldots, y_{14}$ the 14 Legendre nodes on the interval $[-1,1]$ (see (19)). The integrals of (101) – (103) were evaluated at $y_1, y_2, \ldots, y_{14}$, and the relative errors in the l^2

norm were obtained via the formula

$$E_2^{rel} = \frac{\sqrt{\sum_i E^{abs}(y_i)^2}}{\sqrt{\sum_i I(\varphi)(y_i)^2}}, \tag{104}$$

where $E^{abs}(y_i)$ and $I(\varphi)(y_i)$ denote the absolute error and the exact integral (47) evaluated at the point y_i, respectively. The integrals $I(\varphi)(y_i)$ were computed analytically using MATHEMATICA.

In Figure 2, the relative errors of the integrals of (101) - (103) are presented for $N = 6, 8, \ldots, 26$. For comparison, the relative errors of the N-point Gaussian rules (see Theorem 2.9) with $N = 6, 8, \ldots, 26$ applied to the function (100) are shown as well.

Remark 5.1 *The weights (see (72) - (73)) of the quadrature rules (75) - (77) used in Example 5.1 above, depend upon the point of evaluation y. Therefore, for the evaluation of each of the integrals (101) - (103) at each of the points $y_1, y_2, \ldots, y_{14}$, a different set of quadrature weights is used. As an example, in Table 1 we list the quadrature nodes x_n and weights $w_{1,n}$, $w_{2,n}$, $w_{3,n}$ of the* 14*-node version of the quadratures (75) - (77) for the integration of functions with singularities* $\log(|x - y_1|)$, $\frac{1}{y_1 - x}$, $\frac{1}{(y_1-x)^2}$, *with* $y_1 = -0.9862838086968123$ *(the smallest of the 14 Legendre nodes on* $[-1, 1]$*).*

Example 5.2 In this example, we compute the same integrals as in Example 5.1. However, this time we use the quadrature rule (85) that integrates functions of the combined form (81). Specifically, the quadrature weights were constructed via the numerical algorithm described in Section 4 for integrands of the form

$$\sum_{i=1}^{M} \left(a_i + b_i \cdot \log(|y_k - x|) + \frac{c_i}{y_k - x} + \frac{d_i}{(y_k - x)^2} \right) \cdot P_{i-1}(x), \tag{105}$$

for each Legendre node y_k, $k = 1, 2, \ldots, 14$, on the interval $[-1, 1]$ (see (19)). In our computations, we chose the number of weights N equal to $6M$.

In Figure 3 the relative errors (see (104)) are presented for $N = 36, 48, \ldots, 144$.

Example 5.3 In this example, we use the generalized Gaussian quadrature described in Section 3.3 to integrate the functions (101) - (103) where y is a point *outside* but *close* to the interval $[-1, 1]$. Specifically, 36 and 42-node versions of the quadrature formula (97) were constructed for integrands of the form

$$\sum_{i=1}^{M} \left(a_i + b_i \cdot \log(|y - x|) + \frac{c_i}{y - x} + \frac{d_i}{(y - x)^2} \right) \cdot P_{i-1}(x), \tag{106}$$

where $y \in [-10, -1.0016] \cup [1.0016, 10]$. The 36 and 42-node versions were constructed with $M = 11$ and $M = 21$, respectively. In order to test the accuracy of the resulting quadratures, the integrals (101) - (103) were evaluated at 202 equispaced points $y_1, y_2, \ldots, y_{202} \in$

$[-2.002, -1.002] \cup [1.002, 2.002]$, defined by the formula

$$y_k = \begin{cases} -2.002 + 0.01 \cdot (k-1), & \text{for } k = 1, \ldots, 101, \\ 1.002 + 0.01 \cdot (k-102), & \text{for } k = 102, \ldots, 202. \end{cases} \tag{107}$$

In Table 2, the relative errors (see (104)) of the N-point generalized Gaussian quadratures with $N = 36, 42$ applied to the the functions (100), (101) - (103) are presented. For comparison, the relative errors of the N-point Gaussian rules (see Theorem 2.9) with $N = 36, 42, 100, 150, \ldots, 300$ applied to the same functions are shown in Table 3. In Tables 4, 5 we list the quadrature nodes x_n and weights w_n of the 36 and 42-node versions of the quadrature (97).

Example 5.4 In this example, we use a compound quadrature formula based on the combination of the singular quadrature (85), generalized Gaussian quadrature (97), and Gaussian quadrature (see Theorem 2.9) to evaluate the integral

$$F(y) = \text{f.p.} \int_{-1}^{1} \Big(1 + \log(|y-x|) + \frac{1}{y-x} + \frac{1}{(y-x)^2}\Big) \cdot \Big(\sin(200x) + \cos(300x)\Big)\, dx, \tag{108}$$

at several points $y \in (-1, 1)$. Specifically, we subdivide the interval of integration $[-1, 1]$ into K subintervals $I_1, \ldots, I_K$ where

$$I_i = \Big[-1 + \frac{2}{K} \cdot (i-1), -1 + \frac{2}{K} \cdot i\Big], \tag{109}$$

for all $i = 1, 2, \ldots, K$, and then apply a specific quadrature rule on each subinterval to evaluate (108). The quadrature rule used on subinterval I_i is determined by one of the following criteria:

- if $y \in I_i$, then the combined singular quadrature rule (85) is used;
- if $y \notin I_i$ and $y \in I_{i-1} \cup I_{i+1}$, then generalized Gaussian quadrature (97) is used;
- if $y \notin I_i$ and $y \notin I_{i-1} \cup I_{i+1}$, then Gaussian quadrature (see Theorem 2.9) is used.

We denote by

$$y_1^i, y_2^i, \ldots, y_M^i \tag{110}$$

the M Legendre nodes (see (19)) on subinterval I_i. Furthermore, we denote by y_1, y_2, …, y_{MK} the set of all points (110) from all subintervals I_i, $i = 1, 2, \ldots, K$. In other words,

$$y_j^i = y_{M(i-1)+j}, \tag{111}$$

where $i = 1, 2, \ldots, K$ and $j = 1, \ldots, M$. Obviously, by evaluating the integral (108) at the points $y_1, y_2, \ldots, y_{MK}$ via the procedure described above, we obtain approximations to $F(y_1), F(y_2), \ldots, F(y_{MK})$. We perform the calculations with $M = 4, 6, 10, 12, 16$ and $K = 2, 4, 8, \ldots, 8192$; and in order to compare the accuracy for two different choices of

K, we interpolate the obtained values with an M order interpolation scheme to the 100 equispaced points $t_1, t_2, \ldots, t_{100}$ on the interval $(-1, 1)$ defined by the formula

$$t_i = -1 + \frac{2}{101} \cdot i, \tag{112}$$

for all $i = 1, \ldots, 100$.

In Table 6, the relative errors (see (104)) of the scheme described above of degrees $M = 4, 6, 10, 12, 16$ and the number of subintervals $K = 2, 4, 8, \ldots, 8192$, applied to the integral (108) are presented.

The following observations can be made from the examples of this section, and from the more detailed numerical experiments performed by the authors.

1. The quadrature formulae (85), (97) are not convergent in the classical sense; they are only convergent to a prescribed precision ϵ. Needless to say, the two are indistinguishable, as long as the prescribed precision is less than machine precision.
2. The schemes producing the quadrature formulae (75) - (77), (97) do not lose many digits compared to machine precision; constructing the quadratures in double precision arithmetic results in 11 - 12 correct digits; constructing them in extended precision arithmetic results in full double precision accuracy. Needless to say, the nodes and weights of the quadrature formulae (75) - (77), (97) can be (and have been) precomputed and stored, so that the need for extended precision during the *construction* of the quadrature is not a serious limitation.
3. The quadrature formula (85) experiences some loss of precision, not only during the precomputation of the nodes and weights, but also when the formula is applied to specific functions of the form (81). A fairly detailed investigation has led us to the conclusion that the loss of precision is associated with the evaluation of the "hypersingular" function (80), and is unavoidable; the phenomenon is very similar to the loss of precision associated with numerical differentiation, both in character and severity.
4. When the quadrature formulae of this paper are applied to oscillatory functions (of the form (108), or similar), they achieve their full precision at $10 - 15$ nodes per wavelength (for the formulae (75) - (77), (97)), and $20 - 45$ nodes per wavelength (for the formula (85)), respectively.

6 Generalizations and Conclusions

A set of quadratures has been constructed for functions $f : [-1, 1] \to \mathbb{R}$ of the form

$$f(x) = \varphi(x) + \psi(x) \cdot \log|x| + \frac{\eta(x)}{x} + \frac{\theta(x)}{x^2}, \tag{113}$$

where $\varphi, \psi, \eta, \theta : [-1,1] \to \mathbb{R}$ are smooth functions. The term "quadratures" in this case is somewhat of a misnomer, as functions of the form (113) are not integrable in the classical sense, and their integrals are to be interpreted in the appropriate "finite part" sense. One of anticipated applications for such quadratures is the evaluation of integro-pseudo-differential operators (eg. Hilbert transform and derivative of Hilbert transform) arising from the solution of integral equations of potential theory in two dimensions (see, for example, [11, 12]).

The work presented here admits several straightforward extensions:

1. The quadratures in this paper can easily be modified for functions with singularities other than $\log|x|$, $\frac{1}{x}$, $\frac{1}{x^2}$. For example, using Chebyshev polynomials, quadrature formulae similar to (75) – (77), (85) for functions with singularities of the form

$$\frac{\log|x|}{\sqrt{1-x^2}}, \tag{114}$$

$$\frac{1}{x\sqrt{1-x^2}}, \tag{115}$$

$$\frac{1}{x^2\sqrt{1-x^2}}, \tag{116}$$

etc. are easily constructed.

2. A straightforward generalization of the quadratures of this paper in two dimensions leads to quadrature formulae on the square, integrating functions $f : [-1,1]\times[-1,1] \to \mathbb{R}$ of the form

$$f(x_1,x_2) = \varphi(x_1,x_2) + \frac{\psi(x_1,x_2)}{(x_1^2+x_2^2)^{\frac{1}{2}}} + \frac{\eta(x_1,x_2)}{x_1^2+x_2^2} + \frac{\theta(x_1,x_2)}{(x_1^2+x_2^2)^{\frac{3}{2}}}, \tag{117}$$

where $\varphi, \psi, \eta, \theta : [-1,1]\times[-1,1] \to \mathbb{R}$ are smooth functions. Quadrature formulae of this type have been constructed, and the paper reporting them is in preparation.

A Existence of Quadrature Formulae for Functions of the Form $\varphi(x) + \psi(x) \cdot \log|x| + \frac{\eta(x)}{x} + \frac{\theta(x)}{x^2}$

In Section 3.2, we numerically construct quadrature formulae on the interval $[-1,1]$ for functions of the form

$$f(x) = \varphi(x) + \psi(x) \cdot \log|x| + \frac{\eta(x)}{x} + \frac{\theta(x)}{x^2}. \tag{118}$$

The nodes of the quadratures we construct are Gaussian nodes $x_1, x_2, \ldots, x_N$ with a sufficiently large N, and their weights are determined via a least squares procedure. The purpose of this Appendix is to prove that the least squares process of Section 3.2 can be used to obtain quadratures of arbitrary accuracy. We do so by constructing a procedure that, given a real $\epsilon > 0$ and a sufficiently large integer N, produces a set of weights $w_1, w_2, \ldots, w_N$ such that, in combination with the Gaussian nodes $x_1, x_2, \ldots, x_N$ evaluates the integral (82) to precision ϵ.

Remark A.1 *The procedure of this Appendix is quite inefficient, in the sense that it requires a very large number of nodes to obtain acceptable levels of accuracy; its purpose is to prove that such quadratures exist. The procedure for the actual evaluation of coefficients is described in Section 3.2, and results in schemes whose precision is satisfactory at moderate values of N (see Section 5).*

The following lemma follows immediately from the definition of the integral, and the fact that a logarithmic singularity is integrable.

Lemma A.2 *Suppose that $j \geq 0$ is an integer number, and that P_j denotes the j-th Legendre polynomial (see (17)). Then for any positive real number ϵ, there exists an integer $N_0 \geq 1$ such that for any $N \geq N_0$*

$$\left| \int_{-1}^{1} P_j(x) \cdot \log|x| \, dx - \sum_{\substack{i=1 \\ x_i \neq 0}}^{N} w_i \cdot P_j(x_i) \cdot \log|x_i| \right| \leq \epsilon \,, \tag{119}$$

with $x_1, x_2, \ldots, x_N$ and $w_1, w_2, \ldots, w_N$ the nodes and the weights of the N-point Gaussian quadrature (see Theorem 2.9).

The following lemma is an immediate consequence of Lemma A.2.

Lemma A.3 *Suppose that P_j denotes the j-th Legendre polynomial (see (17)). Then for any positive real number ϵ and integer $M \geq 0$, there exists an integer $N_0 \geq 1$ such that for any $N \geq N_0$ and each $j = 0, 1, \ldots, M$*

$$\left| \int_{-1}^{1} P_j(x) \, dx - \sum_{\substack{i=1 \\ x_i \neq 0}}^{N} w_i \cdot P_j(x_i) \right| \leq \epsilon \,, \tag{120}$$

and

$$\left| \int_{-1}^{1} P_j(x) \cdot \log|x| \, dx - \sum_{\substack{i=1 \\ x_i \neq 0}}^{N} w_i \cdot P_j(x_i) \cdot \log|x_i| \right| \leq \epsilon \,, \tag{121}$$

with $x_1, x_2, \ldots, x_N$ and $w_1, w_2, \ldots, w_N$ the nodes and the weights of the N-point Gaussian quadrature (see Theorem 2.9). Furthermore, for any function $F : [-1, 1] \to \mathbb{R}$ of the form

$$F(x) = \sum_{j=0}^{M} \Big(a_j + b_j \cdot \log|x| \Big) \cdot P_j(x) \,, \tag{122}$$

with a_j, b_j arbitrary real coefficients,

$$\left| \int_{-1}^{1} F(x)\,dx - \sum_{\substack{i=1 \\ x_i \neq 0}}^{N} w_i \cdot F(x_i) \right| \leq \epsilon \cdot \sum_{j=0}^{M} \Big(|a_j| + |b_j| \Big) . \tag{123}$$

The following lemma provides a formula for the evaluation of the integrals of functions that are linear combinations of polynomials, and polynomials composed with the singular function $\frac{1}{x^2}$.

Lemma A.4 *Suppose that $n \geq 1$ is an integer number, and that the function $F : [-1, 1] \to \mathbb{R}$ is defined by the formula*

$$F(x) = P_n(x) + \frac{S_n(x)}{x^2} , \tag{124}$$

with $P_n, S_n : [-1, 1] \to \mathbb{R}$ arbitrary polynomials of degree n. Furthermore, suppose that the function $f : [-1, 1] \to \mathbb{R}$ is defined by the formula

$$f(x) = x^2 \cdot F(x) . \tag{125}$$

Then

$$\text{f.p.} \int_{-1}^{1} F(x)\,dx = \sum_{\substack{i=1 \\ x_i \neq 0}}^{n} w_i \cdot \left(F(x_i) - \frac{f(0)}{x_i^2} \right) - 2\,f(0) , \tag{126}$$

where $w_1, w_2, \ldots, w_n$ and $x_1, x_2, \ldots, x_n$ are the weights and nodes of the n-point Gaussian quadrature, respectively (see Theorem 2.9).

Proof. Defining the function $G : [-1, 1] \to \mathbb{R}$ by the formula

$$G(x) = F(x) - \frac{f(0)}{x^2} - \frac{f'(0)}{x} , \tag{127}$$

we observe that G is a polynomial of order n, and therefore

$$\int_{-1}^{1} G(x)\,dx = \sum_{\substack{i=1 \\ x_i \neq 0}}^{n} w_i \cdot \left(F(x_i) - \frac{f(0)}{x_i^2} - \frac{f'(0)}{x_i} \right) . \tag{128}$$

Now, observing that

$$\sum_{\substack{i=1 \\ x_i \neq 0}}^{n} \frac{w_i}{x_i} = 0 , \tag{129}$$

(due to the symmetry of the Gaussian nodes and weights about zero), and substituting (129) into (128), we have

$$\int_{-1}^{1} G(x)\,dx = \sum_{\substack{i=1 \\ x_i \neq 0}}^{n} w_i \cdot \left(F(x_i) - \frac{f(0)}{x_i^2} \right) . \tag{130}$$

It immediately follows from (10) that

$$\text{f.p.} \int_{-1}^{1} \frac{f(0) + f'(0) \cdot x}{x^2} \, dx = -2\, f(0) \,, \tag{131}$$

and, combining (127), (130), (131), we obtain

$$\begin{aligned} \text{f.p.} \int_{-1}^{1} F(x)\, dx &= \int_{-1}^{1} G(x)\, dx + \text{f.p.} \int_{-1}^{1} \frac{f(0) + f'(0) \cdot x}{x^2}\, dx \qquad (132) \\ &= \sum_{\substack{i=1 \\ x_i \neq 0}}^{n} w_i \cdot \left(F(x_i) - \frac{f(0)}{x_i^2} \right) - 2\, f(0) \,. \qquad (133) \end{aligned}$$

□

Lemma A.5 *Suppose that $F : [-1, 1] \to \mathbb{R}$ and $f : [-1, 1] \to \mathbb{R}$ are two functions defined by (124) and (125), respectively. Then there exists a positive real C_1 such that for any sufficiently small h,*

$$\left| f(0) - \Big(F(h) + F(-h) \Big) \cdot \frac{h^2}{2} \right| \leq C_1 \cdot h^2 \,. \tag{134}$$

Furthermore, for any real $\gamma \notin \{-1, 0, 1\}$, there exists a positive real number C_2 such that for any sufficiently small h,

$$\left| f(0) - \Big(F(h) + F(-h) - F(\gamma\, h) - F(-\gamma\, h) \Big) \cdot \frac{\gamma^2 \cdot h^2}{2\, (\gamma^2 - 1)} \right| \leq C_2 \cdot h^4 \,. \tag{135}$$

Proof. We start with observing that for any $F : [-1, 1] \to \mathbb{R}$ defined by (124), there exist such real numbers $a_{-2}, a_{-1}, a_0, a_1, \ldots, a_n$ that

$$F(x) = \frac{a_{-2}}{x^2} + \frac{a_{-1}}{x} + a_0 + a_1\, x + \ldots + a_n\, x^n \,, \tag{136}$$

and due to (125),

$$a_{-2} = f(0) \,. \tag{137}$$

It immediately follows from (136) that for small h,

$$\begin{aligned} F(h) &= \frac{a_{-2}}{h^2} + \frac{a_{-1}}{h} + a_0 + a_1\, h + a_2\, h^2 + O(h^3) \,, \qquad (138) \\ F(-h) &= \frac{a_{-2}}{h^2} - \frac{a_{-1}}{h} + a_0 - a_1\, h + a_2\, h^2 + O(h^3) \,. \qquad (139) \end{aligned}$$

Adding (138) to (139), we obtain

$$F(h) + F(-h) = \frac{2\, a_{-2}}{h^2} + 2\, a_0 + 2\, a_2\, h^2 + O(h^4) \,, \tag{140}$$

and (134) immediately follows from the combination of (137) and (140).

In order to prove (135), we replace h with $\gamma \cdot h$ in (140) above, obtaining

$$F(\gamma h) + F(-\gamma h) = \frac{2\,a_{-2}}{\gamma^2\,h^2} + 2\,a_0 + 2\,a_2\,\gamma^2\,h^2 + O(h^4)\,. \tag{141}$$

Subtracting (141) from (140), we have

$$F(h) + F(-h) - \Big(F(\gamma h) + F(-\gamma h)\Big) = \frac{2\,a_{-2}(\gamma^2 - 1)}{\gamma^2\,h^2} + 2\,a_2\,h^2(1-\gamma^2) + O(h^4)\,, \tag{142}$$

and (135) immediately follows from the combination of (137) and (142). □

The following theorem now immediately follows from the combination of Lemmas A.3 – A.5.

Theorem A.6 *Suppose that P_j denotes the j-th Legendre polynomial (see (17)). Then for any positive real number ϵ and integer $M \geq 0$, there exists an integer $N_0 \geq 1$, real coefficients $\tilde{w}_1, \tilde{w}_2, \ldots, \tilde{w}_N$, and a positive constant C such that for any $N \geq N_0$ and each $j = 0, 1, \ldots, M$*

$$\left| \int_{-1}^{1} P_j(x)\,dx - \sum_{\substack{i=1 \\ x_i \neq 0}}^{N} \tilde{w}_i \cdot P_j(x_i) \right| \leq \epsilon\,, \tag{143}$$

$$\left| \int_{-1}^{1} P_j(x) \cdot \log|x|\,dx - \sum_{\substack{i=1 \\ x_i \neq 0}}^{N} \tilde{w}_i \cdot P_j(x_i) \cdot \log|x_i| \right| \leq \epsilon\,, \tag{144}$$

$$\left| \int_{-1}^{1} \frac{P_j(x)}{x^2}\,dx - \sum_{\substack{i=1 \\ x_i \neq 0}}^{N} \tilde{w}_i \cdot \frac{P_j(x_i)}{x_i^2} \right| \leq \epsilon\,, \tag{145}$$

and

$$\sum_{i=1}^{N} \tilde{w}_i^2 \leq C \cdot \sum_{i=1}^{N} w_i^2\,, \tag{146}$$

with $x_1, x_2, \ldots, x_N$ and $w_1, w_2, \ldots, w_N$ the nodes and the weights of the N-point Gaussian quadrature (see Theorem 2.9). Furthermore, for any function $F : [-1,1] \to \mathbb{R}$ of the form

$$F(x) = \sum_{j=0}^{M} \left(a_j + b_j \cdot \log|x| + \frac{c_j}{x^2} \right) \cdot P_j(x)\,, \tag{147}$$

with a_j, b_j, c_j arbitrary real coefficients,

$$\left| \int_{-1}^{1} F(x)\,dx - \sum_{\substack{i=1 \\ x_i \neq 0}}^{N} \tilde{w}_i \cdot F(x_i) \right| \leq \epsilon \cdot \sum_{j=0}^{M} \Big(|a_j| + |b_j| + |c_j| \Big)\,. \tag{148}$$

x_n	$w_{1,n}$	$w_{2.n}$	$w_{3.n}$
-.9862838086968123E+00	0.6158759029892887E+00	-.1749507584908717E+00	-.1130556007318105E+03
-.9284348836635735E+00	-.3449922634155065E+01	-.2439832523477966E+00	0.2343635742304627E+02
-.8272013150697650E+00	0.6341017949494823E+00	-.2035606965679834E+00	0.2052970256686051E+02
-.6872929048116855E+00	-.1619416300699971E+01	-.2159934769461259E+00	-.1376240462154258E+02
-.5152486363581541E+00	0.4959237125495822E+00	-.1075251867819710E+00	0.1455155616946274E+02
-.3191123689278897E+00	-.1038139679411058E+01	-.1196358314284132E+00	-.1125576720477356E+02
-.1080549487073437E+00	0.3511876142040904E+00	0.1088206509124769E-01	0.1005717417020061E+02
0.1080549487073437E+00	-.6752486832724772E+00	-.1913486054919796E-01	-.7774959517403008E+01
0.3191123689278897E+00	0.2161950693504096E+00	0.9038214134065220E-01	0.6382331406035814E+01
0.5152486363581541E+00	-.4027047889340547E+00	0.4482568706166883E-01	-.4626948764626759E+01
0.6872929048116855E+00	0.1014500308386035E+00	0.1047670461892695E+00	0.3365725647246004E+01
0.8272013150697650E+00	-.1896412777930365E+00	0.5616254115094882E-01	-.2045992407816295E+01
0.9284348836635735E+00	0.2050334687326519E-01	0.6074345322744063E-01	0.1083005671766590E+01
0.9862838086968123E+00	-.3560786461470516E-01	0.2130791084865406E-01	-.2941688960408355E+00

Table 1: 14-node quadratures of the form (75) – (77) for $y = -0.9862838086968123$ (see Example 5.1 and Remark 5.1).

N	1	$(y-x)^{-1}$	$\log(\lvert x-y\rvert)$	$(y-x)^{-2}$
36	0.560E-12	0.250E-13	0.420E-13	0.885E-15
42	0.257E-15	0.119E-14	0.225E-15	0.147E-13

Table 2: Relative errors of the quadrature formula (97) applied to the integrands (100), (101) – (103) (see Example 5.3).

N	1	$(y-x)^{-1}$	$\log(\lvert x-y\rvert)$	$(y-x)^{-2}$
36	0.114E-14	0.581E-02	0.108E-04	0.121E+00
42	0.700E-15	0.277E-02	0.427E-05	0.680E-01
100	0.775E-15	0.192E-05	0.112E-08	0.114E-03
150	0.333E-15	0.350E-08	0.133E-11	0.310E-06
200	0.196E-14	0.631E-11	0.188E-14	0.746E-09
250	0.262E-14	0.106E-13	0.551E-15	0.167E-11
300	0.269E-14	0.967E-15	0.568E-15	0.525E-14

Table 3: Relative errors of the standard Gaussian quadrature (see Theorem 2.9) applied to the integrands (100), (101) – (103) (see Example 5.3).

$\pm x_n$	w_n
0.1065589476527457E+00	0.2116935969670785E+00
0.3113548847160309E+00	0.1954154182193890E+00
0.4932817445063880E+00	0.1668941295018453E+00
0.6431876254991823E+00	0.1325004013421312E+00
0.7584402200373317E+00	0.9850855499442945E-01
0.8418072582807350E+00	0.6923612105413195E-01
0.8991123360358894E+00	0.4649700037042145E-01
0.9369451420922662E+00	0.3015693021568984E-01
0.9611808158857813E+00	0.1907410671190122E-01
0.9763813583671749E+00	0.1186194584542522E-01
0.9857845370872045E+00	0.7299783922072470E-02
0.9915537349540954E+00	0.4465717196444791E-02
0.9950772406715330E+00	0.2722792056317777E-02
0.9972224562334544E+00	0.1654307961017307E-02
0.9985216206191467E+00	0.9963611678876147E-03
0.9992964931862838E+00	0.5843631686022078E-03
0.9997376213125204E+00	0.3153728101867406E-03
0.9999525789657767E+00	0.1230964950065995E-03

Table 4: 36-node generalized Gaussian quadrature (97) for functions of the form (106) with $M = 11$, and precision 10^{-15} (see Example 5.3).

$\pm x_n$	w_n
0.7824400816570354E-01	0.1559838796617961E+00
0.2317400514932991E+00	0.1500543303602524E+00
0.3765817141635966E+00	0.1388302709124357E+00
0.5080234535636137E+00	0.1234870921831402E+00
0.6226938088738944E+00	0.1055618635285824E+00
0.7188418253624399E+00	0.8671614170628514E-01
0.7963343649196293E+00	0.6848351985661966E-01
0.8564163016327517E+00	0.5205731921370713E-01
0.9013001486524265E+00	0.3816842653276627E-01
0.9336896680922276E+00	0.2707608184111357E-01
0.9563457975135937E+00	0.1865610690150748E-01
0.9717714213532305E+00	0.1254153267525754E-01
0.9820411592483684E+00	0.8264234965377917E-02
0.9887573995032291E+00	0.5361830655763248E-02
0.9930900683346067E+00	0.3438177342595994E-02
0.9958561171201172E+00	0.2184437514815405E-02
0.9976063686147585E+00	0.1375256690097983E-02
0.9987019026654443E+00	0.8535706349265051E-03
0.9993734804740140E+00	0.5129451502696074E-03
0.9997640500479557E+00	0.2818251084208615E-03
0.9999571252163234E+00	0.1111565642688685E-03

Table 5: 42-node generalized Gaussian quadrature (97) for functions of the form (106) with $M = 21$, and precision 10^{-15} (see Example 5.3).

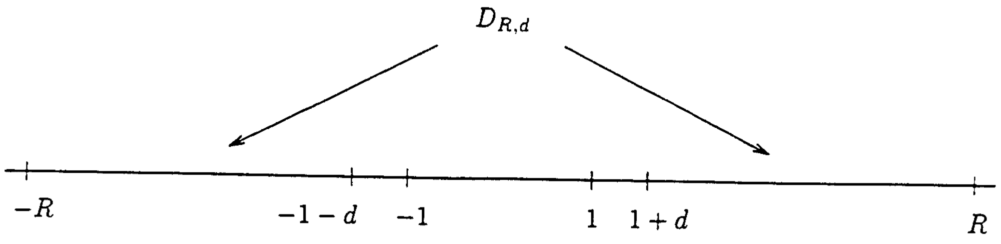


Figure 1: The set $D_{R,d}$.

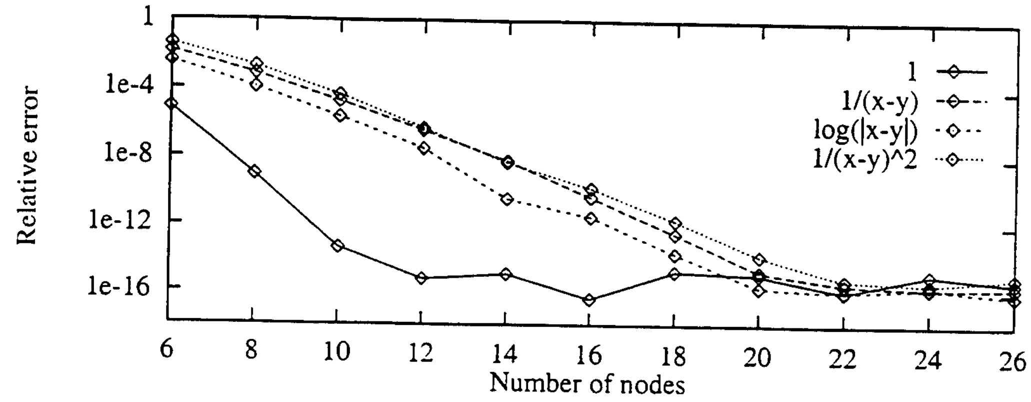


Figure 2: Relative errors of the quadrature formulae (75) - (77) with $N = 6, 8, \dots, 26$ applied to the integrands (101) - (103) (see Example 5.1). The relative error of the N-point Gaussian quadratures with $N = 6, 8, \dots, 26$ applied to the function (100) are presented for comparison.

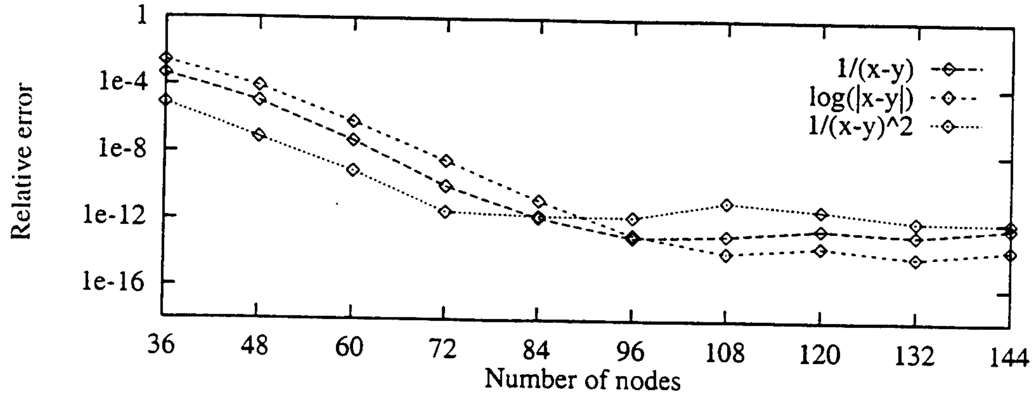


Figure 3: Relative errors of the quadrature formula (85) with $M = 6, 8, \dots, 24$ and $N = 6 \cdot M$, applied to the integrands (101) - (103) (see Example 5.2).

K	degree 4	degree 6	degree 10	degree 12	degree 16
2	0.976E+00	0.105E+01	0.904E+01	0.372E+01	0.799E+01
4	0.109E+01	0.178E+01	0.998E+01	0.622E+01	0.325E+01
8	0.157E+01	0.226E+01	0.429E+01	0.239E+01	0.188E+01
16	0.215E+01	0.149E+01	0.212E+01	0.103E+01	0.788E+00
32	0.131E+01	0.103E+01	0.219E+00	0.483E-01	0.184E-02
64	0.556E+00	0.115E+00	0.194E-02	0.166E-02	0.368E-03
128	0.614E-01	0.285E-02	0.115E-05	0.126E-07	0.364E-09
256	0.442E-02	0.498E-04	0.133E-08	0.270E-09	0.693E-09
512	0.280E-03	0.778E-06	0.837E-09	0.476E-08	0.384E-08
1024	0.165E-04	0.125E-07	0.150E-08	0.149E-07	0.147E-07
2048	0.102E-05	0.271E-08	0.171E-07	0.293E-07	0.532E-07
4096	0.635E-07	0.231E-07	0.613E-07	0.921E-07	0.128E-06
8192	0.110E-07	0.113E-06	0.300E-06	0.134E-05	0.705E-06

Table 6: Relative errors of the compound quadrature formula of degrees $M = 4, 6, 10, 12, 16$ and the number of subintervals $K = 2, 4, \ldots, 8192$ applied to the integral (108) (see Example 5.4).

References

[1] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, vol. 55 of Applied Mathematics Series, Department of Commerce, National Bureau of Standards, 1972.

[2] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen, *LAPACK Users' Guide - Release 2.0*, Society for Industrial and Applied Mathematics, 1994.

[3] A. Björck and G. Dahlquist, *Numerical methods*, Prentice-Hall, Inc., Englewood Cliffs, 1974.

[4] A. J. Burton and G. F. Miller, *The application of integral equation methods to the numerical solution of some exterior boundary-value problems*, Proc. Roy. Soc. Lond. A., 323 (1971), pp. 201–210.

[5] H. Cheng, V. Rokhlin, and N. Yarvin, *Non-linear optimization, quadrature, and interpolation*, Tech. Rep. YALEU/DCS/RR-1169, Computer Science Department, Yale University, 1998.

[6] P. J. Davis and P. Rabinowitz, *Numerical Integration*, Blaisdell Publishing Company, 1967.

[7] D. Gottlieb and S. A. Orszag, *Numerical Analysis of Spectral Methods: Theory and Applications*, Society for Industrial and Applied Mathematics, 6 ed., 1993.

[8] I. S. Gradshteyn and I. M. Ryzhik, *Table of Integrals, Series, and Products*, Academic Press, 5 ed., 1994.

[9] J. Hadamard, *Lectures on the Cauchy's Problem in Linear Partial Differential Equations*, Dover, 1952.

[10] S. Karlin and W. J. Studden, *Tchebycheff Systems: With Applications in Analysis and Statistics*, Interscience Publishers. John Wiley & Sons, 1966.

[11] P. Kolm and V. Rokhlin, *Quadrupole and octuple layer potentials in two dimensions I: Analytical apparatus*, Tech. Rep. Yale YALEU/DCS/RR-1176, Computer Science Department, Yale University, 1999.

[12] P. Kolm and V. Rokhlin, *Quadruple and octuple layer potentials in two dimensions II: Numerical techniques*, in preparation.

[13] M. G. Krein, *The ideas of P. L. Chebyshev and A. A. Markov in the theory of limiting values of integrals*, American Mathematical Society Translations, 12 (1959), pp. 1–122.

[14] J. MA, V. ROKHLIN, AND S. WANDZURA, *Generalized Gaussian quadrature rules for systems of arbitrary functions*, SIAM J. Numer. Anal., 33 (1996), pp. 971–996.

[15] J. R. MAUTZ AND R. F. HARRINGTON, *H-field, E-field, and combined field solutions for conducting bodies of revolution*, AEU, 32 (1978), pp. 157–164.

[16] A. F. PETERSON, *The "interior resonance" problem associated with surface integral equations of electromagnetics: Numerical consequences and a survey of remedies*, Journal of Electromagnetic Waves and Applications, 10 (1990), pp. 293–312.

[17] M. REED AND B. SIMON, *Methods of Modern Mathematical Physics I: Functional Analysis*, Academic Press, 1972.

[18] H. R. SCHWARZ, *Numerical Analysis*, John Wiley, 1989.

[19] J. SONG AND W. C. CHEW, *The Fast Illinois Solver Code: Requirements and Scaling Properties*, IEEE Computational Science and Engineering, (1998), pp. 19–23.

[20] J. STOER AND R. BULIRSCH, *Introduction to Numerical Analysis*, Springer-Verlag, New York, 1993.

[21] G. SZEGÖ, *Orthogonal Polynomials*, vol. 23 of Colloquium Publications, American Mathematical Society, 1939.

[22] N. YARVIN AND V. ROKHLIN, *An improved fast multipole algorithm for potential fields on the line*, SIAM J. Numer. Anal., 36 (1999), pp. 629–666.

A Procedure for the Design of Apparata for the Measurement and Generation of Band-Limited Signals

V. Rokhlin
Research Report YALEU/DCS/RR-1196
March 29, 2000

YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

Whenever physical signals are measured or generated, the locations of receivers or transducers have to be selected. Most of the time, this appears to be done on an ad hoc basis. For example, when a string of geophones is used in the measurements of seismic data in oil exploration, the receivers are located at equispaced points on an interval. When phased array antennae are constructed, their shapes are determined by certain aperture considerations; round and rectangular shapes are common. When antenna beams are steered electronically, it is done by changing the phases (and sometimes, the amplitudes) of the transducers. Again, these transducers are located in a region of predetermined geometry, and their actual locations within that geometry are chosen via some heuristic procedure.
In all these (and many other) cases, the signals being received or generated are *band-limited.* Optimal representation of such signals has been studied in detail by Slepian et. al. more than 30 years ago, and some of the obtained results were applied by D. Rhodes to the design of antenna patterns; further development of this line of research appears to have been hindered by the absence at the time of necessary numerical tools. We combine these classical results with the recently developed apparatus of Generalized Gaussian Quadratures to construct optimal nodes for the measurement and generation of band-limited signals. In this report, we describe the procedure based on these techniques for the design of such receiver (and transducer) configurations in a variety of environments.

A Procedure for the Design of Apparata for the Measurement and Generation of Band-Limited Signals

V. Rokhlin
Research Report YALEU/DCS/RR-1196
March 29, 2000

The author was supported in part by DARPA/AFOSR under Contract F49620/97/1/0011, in part by ONR under Grant N00014-96-1-0188, an in part by AFOSR under STTR number F49620/98/C/0051
Approved for public release: distribution is unlimited
Keywords: *Band-limited Signals, Antenna Arrays, Beam-forming*

1 Introduction

When measurements are performed, it often happens that the signal to be measured is well approximated by linear combinations of oscillatory exponentials, i.e. functions of the form

$$\sum_{j=1}^{n} a_j \cdot e^{i \cdot \lambda_j \cdot x} \tag{1}$$

in one dimension, of the form

$$\sum_{j=1}^{n} a_j \cdot e^{i \cdot (\lambda_j \cdot x + \mu_j \cdot y)} \tag{2}$$

in two dimensions, and of the form

$$\sum_{j=1}^{n} a_j \cdot e^{i \cdot (\lambda_j \cdot x + \mu_j \cdot y + \nu_j \cdot z)} \tag{3}$$

in three dimensions. In most cases, the signal is band-limited, i.e. there exist such real positive a that all $1 \leq j \leq n$.

$$\mid \lambda_j \mid \leq a \tag{4}$$

in one dimension,

$$\lambda_j^2 + \mu_j^2 \leq a^2 \tag{5}$$

in two dimensions, and

$$\lambda_j^2 + \mu_j^2 + \nu^2 \leq a^2, \tag{6}$$

in three dimensions.

As is well-known, most measurements of electromagnetic and acoustic data (especially at reasonably high frequencies) are of this form. Examples of such situations include geophone and hydrophone strings in geophysics, phased array antennae in radar systems, multiple transceivers in ultrasound imaging, and a number of other applications in astrophysics, medical imaging, non-destructive testing, etc.

In this report, we describe a procedure for determining the optimal distribution of sources and receivers that maximizes accuracy and resolution in measuring band-limited data given a fixed number of receivers. Alternatively, the procedure can be used to determine the optimal distribution of receivers that will minimize their number given specified accuracy and resolution. While the techniques described in this note are fairly general, we describe them in detail in the case of linear antenna arrays; the changes needed to generalize the approach to other cases are summarized in Section 6.

Remark 1.1 One of principal issues in the design of antenna arrays is the treatment (or avoidance) of the so-called supergain (or superdirectivity). Supergain is the condition that occurs when an antenna design is attempted that is prohibited (or nearly prohibited) by the Heisenberg principle; technically, it occurs in the form of very closely spaced elements operating out of phaze, and leads to prohibitive Ohmic losses in transmitting antennae, loss of sensitivity in receiving ones, etc. Since the purpose of this note is to introduce techniques for selecting the locations of elements *for a prescribed antenna pattern*, we avoid the issue of choosing the antenna pattern altogether. Instead, we observe design optimal element distributions for several standard far-field patterns (see Section 5.1), and we observe that the scheme for choosing optimal distributions of elements is virtually independent of the patterns being approximated.

Technically, the approach taken here is to observe that designing an antenna array can be viewed as constructing a quadrature formula for the integration of certain special classes of functions. Using recently developed techniques for the construction of so-called Generalized Gaussian Quadratures, we obtain both nodes and weights that are optimal (in a very strong sense) for the required antenna pattern.

The structure of this note is as follows. In Section 2, we summarize some of the mathematical apparatus to be used: Chebychev Systems, Generalized Gaussian Quadratures, etc. In Section 3, we recapitulate some of the standard antenna theory, primarily to introduce the necessary notation. In Section 4, element distributions given a specific antenna pattern. In Section 5, we illustrate our approach with several numerical examples, and Section 6 contains a discussion of the generality of the schemes presented.

2 Analytical Preliminaries

In this section, we summarize several known facts about classical Special functions. All of these facts can be found in the literature; detailed references are given in the text.

2.1 Chebyshev systems

Definition 2.1 *A sequence of functions $\phi_1, \ldots, \phi_n$ will be referred to as a* Chebyshev system *on the interval $[a, b]$ if each of them is continuous and the determinant*

$$\begin{vmatrix} \phi_1(x_1) & \cdots & \phi_1(x_n) \\ \vdots & & \vdots \\ \phi_n(x_1) & \cdots & \phi_n(x_n) \end{vmatrix} \tag{7}$$

is nonzero for any sequence of points $x_1, \ldots, x_n$ such that $a \leq x_1 < x_2 \ldots < x_n \leq b$.

An alternate definition of a Chebyshev system is that any linear combination of the functions with nonzero coefficients must have no more than n zeros.

Examples of Chebyshev and extended Chebyshev systems include the following (additional examples can be found in [8]).

Example 2.1 *The powers $1, x, x^2, \ldots, x^n$ form an extended Chebyshev system on the interval $(-\infty, \infty)$.*

Example 2.2 *The exponentials $e^{-\lambda_1 x}, e^{-\lambda_2 x}, \ldots, e^{-\lambda_n x}$ form an extended Chebyshev system for any $\lambda_1, \ldots, \lambda_n > 0$ on the interval $[0, \infty)$.*

Example 2.3 *The functions $1, \cos x, \sin x, \cos 2x, \sin 2x, \ldots, \cos nx, \sin nx$ form a Chebyshev system on the interval $[0, 2\pi]$.*

Example 2.4 *Suppose that $c > 0$ is a real number, w is a positive function $[-1,1] \to \mathbb{R}$ such that $w \in c^1[-1,1]$ and $w(-x) = w(x)$ for all $x \in [-1,1]$, n is a natural number, and the operators $P, Q : L^2[-1,1] \to L^2[-1,1]$ are defined by the formulae*

$$P(\phi)(x) = \int_{-1}^{1} w(t) \cdot e^{i \cdot c \cdot x \cdot t} \cdot \phi(t) \; dt \tag{8}$$

$$Q = P^* \circ P. \tag{9}$$

Suppose further that $\phi_1, \phi_2, \ldots$ are the eigenfunctions of Q, $\lambda_1, \lambda_2, \ldots$ are the corresponding eigenvalues, and $\lambda_1 > \lambda_2 > \lambda_3 \ldots$. Then all eigenfunctions of Q (also known as the right singular vectors of P) can be chosen to be real. Furthermore, the functions $\phi_1, \phi_2, \ldots, \phi_n$ constitute a Chebychev system on the interval $[-1,1]$.

2.2 Generalized Gaussian quadratures

A quadrature rule is an expression of the form

$$\sum_{j=1}^{n} w_j \cdot \phi(x_j), \tag{10}$$

where the points $x_j \in \mathbb{R}$ and coefficients $w_j \in \mathbb{R}$ are referred to as the nodes and weights of the quadrature, respectively. They serve as approximations to integrals of the form

$$\int_a^b \phi(x) \cdot \omega(x) dx \tag{11}$$

with ω is an integrable non-negative function.

Quadratures are typically chosen so that the quadrature (10) is equal to the desired integral (11) for some set of functions, commonly polynomials of some fixed order. Of these, the classical Gaussian quadrature rules consist of n nodes and integrate polynomials of order $2n - 1$ exactly. In [13], the notion of a Gaussian quadrature was generalized as follows:

Definition 2.2 *A quadrature formula will be referred to as* Gaussian *with respect to a set of $2n$ functions $\phi_1, \ldots, \phi_{2n} : [a,b] \to \mathbb{R}$ and a weight function $\omega : [a,b] \to \mathbb{R}^+$, if it consists of n weights and nodes, and integrates the functions ϕ_i exactly with the weight function ω for all $i = 1, \ldots, 2n$. The weights and nodes of a Gaussian quadrature will be referred to as Gaussian weights and nodes respectively.*

The following theorem appears to be due to Markov [15, 16]; proofs of it can also be found in [10] and [8] (in a somewhat different form).

Theorem 2.1 *Suppose that the functions* $\phi_1, \ldots, \phi_{2n} : [a, b] \to \mathrm{R}$ *form a Chebyshev system on* $[a, b]$. *Suppose in addition that* $\omega : [a, b] \to \mathrm{R}$ *is a non-negative integrable function* $[a, b] \to \mathrm{R}$. *Then there exists a unique Gaussian quadrature for the functions* $\phi_1, \ldots, \phi_{2n}$ *on* $[a, b]$ *with respect to the weight function* ω. *The weights of this quadrature are positive.*

Remark 2.1 While the existence of Generalized Gaussian Quadratures was observed more than 100 years ago, the constructions found in [15, 16], [3, 10], [7, 8] do not easily yield numerical algorithms for the design of such quadrature formulae; such algorithms have been constructed recently (see [13, 28, 2]). The version of the procedure found in [2] was used to produce the results presented in the Examples 5.1, 5.2, 5.3 in Section 5.1; the reader is referred to [2] for details.

Applying Theorem 2.1 to the Example 2.4, we obtain the following theorem.

Theorem 2.2 *Suppose that under the conditions of Example 2.4, n is even. Then there exist* $n/2$ *points* $t_1, t_2, \ldots, t_{n/2}$ *on the interval* $[-1, 1]$ *and positive real numbers* $w_1, w_2, \ldots, w_{n/2}$ *such that*

$$\int_{-1}^{1} w(t) \cdot \phi_i(t) \; dt = \sum_{j=1}^{n/2} w_j \cdot \phi_i(t_j), \tag{12}$$

for all $i = 1, 2, \ldots, n$, *with* $\phi_1, \phi_2, \ldots, \phi_n$ *the first* n *eigenfunctions of the operator* Q *defined in (9).*

Corollary 2.3 *The above theorem provides a tool for the efficient approximate evaluation of integrals of the form (12), as follows. Given a positive real* ϵ, *we construct the*

Singular Value Decomposition of the operator P defined in (8). Choosing n to be the smallest even integer such that

$$\sum_{j=n+1}^{\infty} \lambda_j^2 < \epsilon^2, \tag{13}$$

we construct an $n/2$-point quadrature that integrates n first right singular functions exactly (effective numerical schemes for the construction of such quadratures can be found in [13, 28, 2]). Now, we observe that due to the triangle inequality combined with the positivity of the obtained weights $w_1, w_2, \ldots, w_{n/2}$,

$$|\sum_{j=1}^{n/2} w_j \cdot e^{i\cdot c\cdot x\cdot t_j} - \int_{-1}^{1} w(x) \cdot e^{i\cdot c\cdot x\cdot t}\, dt| < \epsilon \tag{14}$$

for any $x \in [-1, 1]$.

Remark 2.2 The principal subject of this note is the fact that the pattern of an antenna array is formed by a physical process amounting to a hardware implementation of a quadrature formula for functions of the form (9). Thus, designing a configuration of elements for such an antenna is equivalent to constructing a quadrature formula for functions of the form(9), and can be achieved via the techniques described in [13, 28, 2]).

3 Elements of Antenna Theory

In this section, we summarize certain facts about the theory of linear antenna arrays; all of these facts are well-known, and can be found, for example, in [9].

3.1 Pattern of a linear array

A source distribution σ on the interval $[-1, 1]$ creates the far-field pattern $f : [0, \pi] \to \mathbb{C}$ given by the formula

$$f(\theta) = \int_{-1}^{1} \sigma(u) \cdot e^{i\cdot k\cdot u\cdot cos(\theta)}\, du, \tag{15}$$

where k is the free-space wavenumber, u is the point on the interval $[-1,1]$, and θ is the angle between the point on the horizon where the far field is being evaluated and the x-axis. It is customary to introduce the notation

$$x = cos(\theta), \tag{16}$$

and define the function $F : [-1,1] \to \mathbb{C}$ by the formula

$$F(x) = f(acos(x)). \tag{17}$$

Now, defining the operator $A : L^2[-1,1] \to L^2[-1,1]$ by the formula

$$A(\sigma)(x) = \int_{-1}^{1} \sigma(u) \cdot e^{i \cdot k \cdot u \cdot x}\, du, \tag{18}$$

we observe that

$$F = A(\sigma) = \int_{-1}^{1} \sigma(u) \cdot e^{i \cdot k \cdot u \cdot x}\, du. \tag{19}$$

The function F is usually more convenient to work with than f, and the following obvious lemma is the principal reason for this difference.

Lemma 3.1 *Suppose that $\sigma \in L^2[-1,1]$, the function $F \in L^2[-1,1]$ is defined by (19), α is a real number, and the function $\tilde{\sigma} \in L^2[-1,1]$ is defined by the formula*

$$\tilde{\sigma}(u) = e^{i \cdot \alpha \cdot u} \cdot \sigma(u). \tag{20}$$

Then

$$A(\tilde{\sigma})(x) = A(\sigma)(x - \alpha) \tag{21}$$

for all $x \in (-\infty, \infty)$. In other words, in order to translate the antenna pattern F (viewed as a function of $x = cos(\theta)$) by α, one has to multiply by $e^{i \cdot \alpha \cdot k}$ the source distribution σ generating the pattern F.

Observation 3.1 *While the obvious physical considerations lead to the antenna pattern F defined on the interval $[-1, 1]$, the formulae (15), (17) also define naturally the extension of F to the function $\mathbb{R} \rightarrow \mathbb{C}$; in a mild abuse of notation, we will be denoting by F both the original mapping $[-1, 1] \rightarrow \mathbb{C}$ and its extension to the mapping $\mathbb{R} \rightarrow \mathbb{C}$. Similarly, we will be denoting by A both the operator $L^2[-1, 1] \rightarrow L^2[-1, 1]$ defined by (18) and its natural extension mapping $L^2[-1, 1] \rightarrow c^\infty(\mathbb{R})$. The restriction of F on $\mathbb{R} \backslash [-1, 1]$ is referred to as the invisible spectrum of the source distribution σ and plays an important role in the antenna theory (this role is discussed briefly in the following subsection). By the same token, the restriction of F on the interval $[-1, 1]$ is referred to as the visible spectrum.*

When an antenna array is implemented in hardware, it is (usually) constructed of a finite collection of elements, as opposed to being a continuous source distribution. Mathematically, it is equivalent to replacing the general function σ in (15), (19) with σ defined by the expression

$$\sigma(x) = \sum_{j=1}^{n} \beta_j \cdot \phi_j(u), \tag{22}$$

with $\phi_1, \phi_2, \ldots, \phi_n$ the source distributions generated by individual elements, and the coefficients $\beta_1, \beta_2, \ldots, \beta_n$ the intensities of the elements. As a rule, the elements are localized in space (i.e. the functions $\phi_1, \phi_2, \ldots, \phi_n$ are supported on small subintervals of $[-1, 1]$), and very often, all of the elements are identical (i.e. the functions ϕ_j are translates of each other), so that

$$\phi_j(u) = \phi(u - u_j), \tag{23}$$

with ϕ the source distribution of a single element located at the point $u = 0$, and u_j the location of the element number j. Obviously, the far-field pattern of ϕ is given by the formula

$$F_\phi(x) = \int_{-1}^{1} \phi(u) \cdot e^{i \cdot k \cdot u \cdot x} \, du; \tag{24}$$

combining (24) with (22) and (23), we obtain the identity

$$\sigma(x) = \int_{-1}^{1} \phi(u) \cdot e^{i \cdot k \cdot u \cdot x} \, du \cdot \sum_{j=1}^{n} \beta_j \cdot e^{i \cdot k \cdot u_j \cdot x}, \tag{25}$$

known in the antenna theory as the principle of pattern multiplication.

Remark 3.2 The standard form of the principle of multiplication reads: "The field pattern of an array of nonisotropic but similar point sources is the product of the pattern of the individual source and the the pattern of an array of isotropic point sources, having the same locations, relative amplitudes and phases as the nonisotropic point sources" (see [9]). Needless to say, this is a special case of the well-known theorem from the theory of the Fourier Transform, stating that the Fourier transform of the product of two functions is the convolution of the Fourier Transforms of multiplicants.

4 Antenna Patterns and Corresponding Optimal Element Distributions

4.1 Characteristics of an antenna pattern

Depending on the situation, the design of an antenna array attempts to optimize certain characteristics of the resulting far-field pattern, subject to certain constraints on the number, power, etc. of the elements. Since the principal purpose of this note is to describe a technique for the selection of the *locations* of the elements that approximate a user-specified pattern, we could use any reasonable far-field pattern to be approximated. In subsection 4.2, 4.3, we construct optimal element distributions for the so-called sector patterns and cosecant pattern, respectively; a detailed discussion of these (and several other) pattern cans be found, for example in [14].

We will say that the antenna pattern has the ϵ-bandwidth b if

$$\int_{b \le ||x|| \le 1} |F(x)|^2 \, dx = \epsilon^2 \cdot \int_{-1}^{1} |F(x)|^2 \, dx \tag{26}$$

in other words, the proportion of the energy radiated outside the ϵ-beamwidth from the axis of the beam is equal to ϵ. The *supergain* of an antenna is defined (see, for example, [27]), as the ratio

$$\frac{\int_{-\infty}^{+\infty} |F(x)|^2 \, dx}{\int_{-1}^{1} |F(x)|^2 \, dx}. \tag{27}$$

The supergain (sometimes referred to as superdirectivity) measures the ratio of the energy associated with the total spectrum of the antenna to the energy in its visible spectrum; while detailed discussion of supergain and related issues is outside the scope of this note, we will observe that antenna arrays with large degrees of supergain would violate the uncertainty principle, and thus are physically impossible. Attempts to construct supergain antennae result in rapidly (exponentially) growing Ohmic losses, prohibitive accuracy requirements, extremely low bandwidth, etc. Thus, any potentially useful procedure for the design of antenna arrays has to limit the supergain of the resulting patterns.

4.2 Sector patterns

It is often desirable to construct antenna patterns that are as constant as possible within the main beam, and as small as possible outside it; in other words, ideally, the pattern would be defined by the formulae

$$F_b(x) = 1 \quad for \quad |x| \le b, \tag{28}$$

$$F_b(x) = 0 \quad for \quad |x| > b, \tag{29}$$

with b a real number such that $0 < b \le k$. Needless to say, the function F_b defined by the formulae (28), (29) is not band-limited, and some approximation has to be used. A standard procedure is to truncate the Fourier Transform of F_b, approximating it by the function $\tilde{F}_b$ defined by the formula

$$\tilde{F}_b(x) = \int_{-1}^{1} \frac{sin(b \cdot t)}{t} \cdot e^{i \cdot k \cdot x \cdot t} \tag{30}$$

(see, for example, [26]). An important special case occurs when $b = k$, with (30) assuming the form

$$\tilde{F}_k(x) = \int_{-1}^{1} \frac{sin(k \cdot t)}{t} \cdot e^{i \cdot k \cdot x \cdot t}; \tag{31}$$

obviously, the latter expression is a band-limited approximation of the δ-function. Another frequently encountered situation is that of $b = k/2$, so that (30) assumes the form

$$\tilde{F}_k(x) = \int_{-1}^{1} \frac{sin(\frac{k}{2} \cdot t)}{t} \cdot e^{i \cdot k \cdot x \cdot t}. \tag{32}$$

which is a band-limited approximation to the beam that is equal to 1 for $-1/2 < x < 1/2$ and to zero elsewhere.

In Section 4.4 below, we demonstrate optimal element configurations that produce approximations to the patterns (31), (32) with $k = 20\pi, 10\pi, 32.4676\pi$.

Remark 4.1 While (30) is by no means the only possible band-limited approximations to to F_b, it is quite satisfactory in most cases, in addition to being simple. Furthermore, the principal purpose of this note is to describe a technique for the selection of *locations* of the nodes, given a pattern to be approximated. Thus, we ignore the issue of the optimal choice of F_b.

4.3 Cosecant patterns

Another standard far-field radiation pattern is the so-called cosecant pattern (see, for example, [19]). Given two real numbers $0 < a < b < 1$, the cosecant pattern $F_{a,b}$ is defined by the formula

$$F_{a,b}(x) = \frac{1}{x} \tag{33}$$

for all $x \in [a, b]$, and

$$F_{a,b}(x) = 0 \tag{34}$$

for all $x \in ([-1,1] \setminus [a,b])$. Again, the function $F_{a,b}$ defined by the formulae (33), (34) is not band-limited, and can not be represented by the expression of the form (24). Before the scheme of this note can be applied to $F_{a,b}$, the latter has to be approximated with a band-limited function; as discussed in Section 4.1 above, if such an approximation is to be useful as an antenna pattern, its supergain factor has to be controlled. Fortunately, a procedure for such an approximation has been in existence for more than 35 years (see, [18]); the algorithm of [18] is a modification of the least-squares approach *permitting the user to limit the supergain factor of the obtained pattern explicitly.* At the time, the utility of the scheme of [18] was limited by the (perceived) difficulty in the numerical evaluation of Prolate Spheroidal Wave functions; given the present state of numerical analysis, this difficulty is non-existent, and it is this author's impression that the insights of [18], [19] deserve more attention than they have been receiving.

4.4 Optimal distributions of elements

In this subsection, we briefly describe an algorithm for the construction of optimal (in the sense defined below) element configurations for the generation of antenna patterns given by (15), of which the patterns (29)-(31) are special cases. As will be seen, the procedure is in fact applicable to the design of element configurations for very general far-field patterns.

We start with observing that (15) expresses the far-field pattern F as an integral over the interval $[-1,1]$ of functions of the form

$$\sigma(u) \cdot e^{i \cdot k \cdot x \cdot u}, \tag{35}$$

with $x = cos(\theta)$ determined by the direction θ in which the far-field is being evaluated. In other words, the problem of finding efficient antenna element distributions is equivalent to that of constructing quadrature formulae for integrals of the form (8), with

$$w(t) = \sigma(t). \tag{36}$$

In the cases when σ is non-negative everywhere on the interval $[-1,1]$, Theorem 2.2 guarantees the existence of Generalized Gaussian Quadratures, and [13, 28]) provide a satisfactory numerical apparatus for the construction of such quadratures. Obviously, the patterns given by the formula (28) are not generated by non-negative source distributions, except when

$$b \leq \pi. \tag{37}$$

Thus, for these (and many other) patterns, the conditions of Theorem 2.2 are violated, and the existence of Generalized Gaussian Quadratures is not guaranteed. In our numerical experiments, the techniques of [2]) (after some tuning) have always been successful in finding the Gaussian quadratures for integrals of the form (28); some of our results are presented in Section 5 below.

5 Numerical Examples

In this section, we present examples of optimal element distributions generating the patterns of the preceding Section; all of the results presented here have been obtained numerically. Antenna patterns we present are compared to the antenna patterns given by uniform source distributions; configurations of elements approximating these antenna patterns are compared to equispaced distributions of elements generating the same antenna patterns.

5.1 Optimal distributions of elements

In this section, we demonstrate the results of the application of the techniques of Section 4.4 of this note to the types of antenna patterns described in the Sections 4.2, 4.3.

In all cases, we choose the size of an antenna array and a pattern to be reproduced, and use the scheme outlined in Section 4.4 to design a distribution of antenna elements (both the locations and the intensities) located within the chosen array that reproduces the required pattern. For comparison, we also generate optimal (in the least squares sense) approximations to the desired pattern generated by equispaced elements located within the same array. Since the number of equispaced nodes required to obtain a reasonable approximation to the desired pattern is (in many cases) much greater than the number of optimally chosen nodes, for each example we demonstrate patterns generated by several such configurations. In this manner, the numbers of optimally chosen nodes necessary to obtain reasonable approximations to the desired patterns can be compared to the numbers of equispaced nodes required to obtain similar results.

5.1.1 Sector patterns

Example 5.1 *The first example we consider is of the pattern defined by the formula (32), with* $k = 62.8312$, *so that the size of the array is* 20 *wavelengths.*

In Figure 5, we display an approximation to the pattern obtained with 19 *elements, overlayed with the exact pattern; the locations of the elements are displayed in Figure 5a; the relative error of the obtained approximation is* 5.01%.

Similarly, in Figure 5g, we display the approximation to the pattern obtained with 21 *elements, overlayed with the exact pattern; the relative error of the obtained approximation is* 0.443%; *in Figure 5h, we display the the approximation obtained with* 17 *elements. In the latter case, the relative error of the obtained approximation is* 6.43%; *Figure 5i depicts the* 17*-node distribution producing the approximation illustrated in Figure 5h. Finally, Figure 5j contains a graph of the values of the sources located at the* 17 *nodes depicted in Figure 5i and generating the pattern shown in Figure 5h.*

For comparison, the optimal approximation obtained with 19, 24, 29, 31, *and* 34 *equispaced elements are displayed in Figures 5b, 5c, 5d, 5e, 5f, respectively; these are also overlayed with the exact pattern.*

Example 5.2 *Our second example is identical to the first one, with the exception that* $k = 31.416$, *so that the size of the array is* 10 *wavelengths.*

In Figure 6, we display an approximation to the pattern obtained with 9 *elements, overlayed with the exact pattern; the locations of the elements are displayed in Figure 6a; the relative error of the obtained approximation is* 11.2%.

Similarly, in Figure 6f, we display the approximation to the pattern obtained with 11 *elements, overlayed with the exact pattern; the relative error of the obtained approximation is* 0.600%.

For comparison, the optimal approximation obtained with 9, 14, 16, *and* 18 *equispaced elements are displayed in Figures 6b, 6c, 6d, 5e, respectively; these are also overlayed with the exact pattern.*

Example 5.3 *Our third example is identical to the preceding two, with the exception that* $k = 102$, *so that the size of the array is about* 32.45 *wavelengths.*

In Figure 7a, we display an approximation to the pattern obtained with 23 *optimally distributed elements, overlayed with the exact pattern and with the pattern obtained with* 23 *equispaced elements.*

The relative error of the obtained approximation is 5.4%; *needless to say, the error of the approximation obtained with the equispaced nodes is more than* 70%. *As can be seen from Figure 7c, the actual size of the obtained* 23*-element array is about* 21 *wavelengths; in other words, in order to obtain this precision, the array needs to be about* 2/3 *of the nominal (maximum permitted) length.*

In Figure 7b, we display the approximation to the pattern obtained with 42 *and* 48 *elements, overlayed with the exact pattern.*

It is worth noting that with 33 *optimally distributed elements, the pattern is approximated to the precision* 0.12%; *we do not display the obtained pattern since it is visually indistinguishable from the pattern being approximated.*

Example 5.4 *Our final example is somewhat different from the preceding ones, in that instead of approximating a sector pattern, we approximate a cosecant pattern (see (33), (34) in Subsection 4.3 above).*

In this example, we set

$$a = sin(15°), \tag{38}$$

$$b = sin(75^\circ), \tag{39}$$

and use the procedure of [18] to approximate $F_{a,b}$ with a band-limited function. The band-limit has been more or less arbitrarily set to 110, *resulting in an antenna array about* 35 *wavelengths in size, and the supergain factor of the approximation was set to* 1.1.

In Figure 8a, we display an approximation to the pattern obtained with 53 *optimally distributed elements, overlayed with the exact bandlimited pattern and with the pattern obtained with* 53 *equispaced elements.*

The relative error of the obtained approximation is 1.79%*; the error of the approximation obtained with the equispaced nodes is about* 42%.

In Figure 8b, we display the approximation to the pattern obtained with 47 *optimally distributed elements, overlayed with the exact pattern; the purpose of this final figure is to demonstrate the behavior of the scheme when the number of elements is insufficient (i.e. when the array is underresolved).*

It is worth noting that it takes about 70 *equispaced nodes to obtain the resolution obtained with* 47 *optimally chosen ones.*

The following observations can be made from Figures 5 - 8b, and from the more detailed numerical experiments performed by the author.

1. In order to obtain reasonable precision, the scheme requires about 1 point per wavelength in the antenna array; this is more or less independent from the structure of the beam *as long as the pattern is symmetric about the point $x = 0$*. This fact is observed numerically, even for modest numbers of nodes; for large-scale arrays, this statement (interpreted asymptotically) can be proved rigorously. For certain beam structures, the required number of nodes is even less (see Example 5.3). The reasons for these additional savings are subtle, and have to do with the fact that the continuous source distribution generating the pattern is relatively small on a large part of the antenna array; the algorithm of [2] takes advantage of this fact to reduce the number of nodes. When the beam is not symmetric about $x = 0$, the number of elements required does depend on

the structure of the pattern, and the dependence is fairly complicated. Generally, the improvement for non-symmetric beams is less than that for the symmetric ones.

2. The qualiative behavior of the scheme is similar to that of the Gaussian quadratures in that it displays no convergence at all until a certain minimum number of nodes is achieved; after that, the convergence is very fast. This behavior is not surprising, since the scheme *is* based on a Generalized Gaussian quadrature.

3. For the sector pattern with the sector $[-1/2, 1/2]$, the scheme reduces the required number of nodes by a factor of about 1.5 for small-scale problems, and roughly by a factor of 2 for large-scale ones; again, for large-scale problems, an asymptotic version of this statement can be proven rigorously.

4. For the cosecant pattern with the parameters specified by (38), (39), the number of nodes required is reduced by approximately a factor of 1.4. As the sidelobe level is reduced, the improvement obtained by going from the equispaced discretization to the optimal one increases rapidly.

5. An examination of Figures 5a, 6a shows that while the optimal nodes are by no means uniform, they display no clustering behavior.

6. An examination of Figure 5j shows that the intensities of individual elements do not become large; this is confirmed by the more extensive numerical experiments performed by the author.

7. The combination of the preceding two paragraphs (combined with additional numerical experiments and analysis) provide evidence that configurations of this type should pose no supergain problems.

6 Generalizations

The results described above admit radical generalizations in several directions; several such directions are discussed below,

1. Conformal one-dimensional arrays. The extension of the techniques of this note to one-dimensional arrays located on curves in R^3 is completely straightforward, involving only a modest increase of the CPU time requirements of the procedure. Improvement in the number of nodes required to produce a prescribed pattern is similar to that in the case of a linear array.

2. Planar two-dimensional arrays. A straightforward generalization of the results of Sections 4, 5, is to rectangular planar arrays. Here, a tensor product quadrature can be constructed from the quadratures of Sections 4, 5, possessing all of the desirable properties of the latter. Obviously, the advantage in the number of transducers is squared, so that (for example) replacing 50 nodes in each of the two directions by 23 nodes (see Example 5.3 above) will lead to a factor of $(50/23)^2 \sim 4.7$ savings in the number of elements.

The theory of Section 4 has been extended for disk-shaped arrays, via (*inter alia*) the techniques developed in [23]. The improvement in the number of nodes is comparable to that obtained in the rectangular geometry, and the CPU time requirements do not differ appreciably from those in the case of linear one-dimensional arrays.

The extension of the theory to more general geometries in the plane is in progress. At the present time, our only numerical experiments have been with arrays on triangles; the results are encouraging, but the CPU time requirements of the algorithms are excessive (we have only been able to design triangular arrays about 6 wavelengths in size). We are now in the process of constructing a more efficient numerical procedure for such computations.

3. Conformal two-dimensional arrays. The only environment in which we have a satisfactory theory is when the array is located on a surface of revolution; even in this environment, no experiments have been performed. We have not investigated more general conformal two-dimensional arrays in sufficient detail.

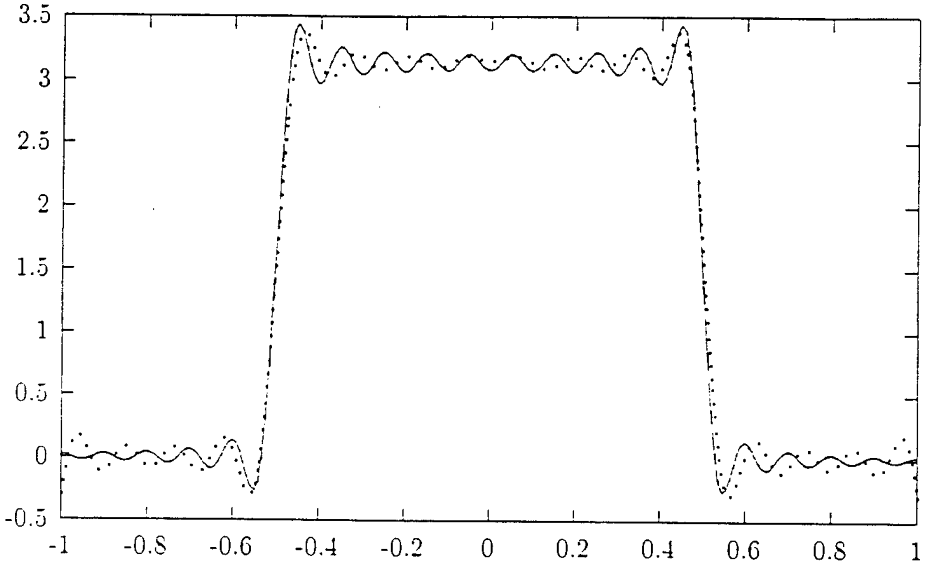


Figure 5: The pattern created by the 19 optimal elements. depicted in Figure 5a as described in Example 5.1

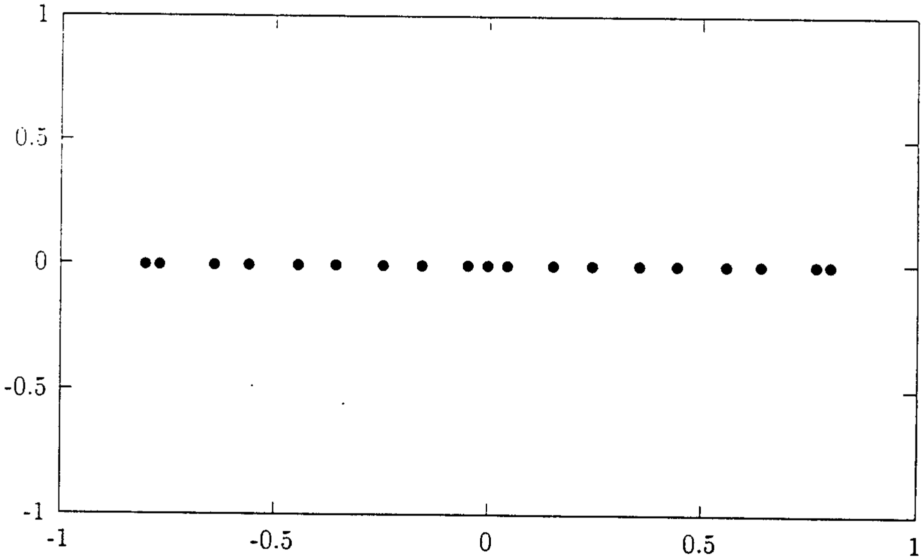


Figure 5a: The distribution of elements creating the pattern depicted in Figure 5, as described in Example 5.1

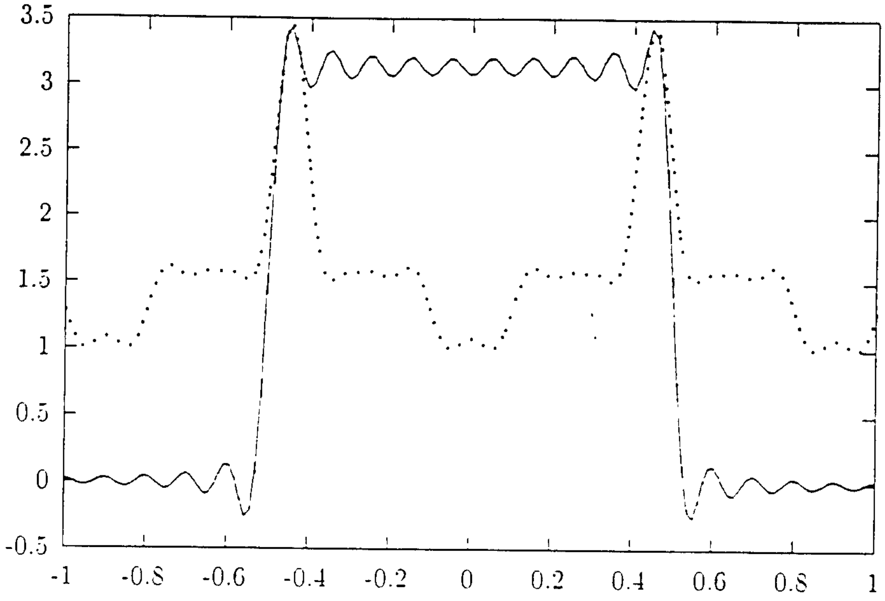


Figure 5b: The optimal approximation to the sector pattern generated by 19 equispaced nodes, as described in Example 5.1

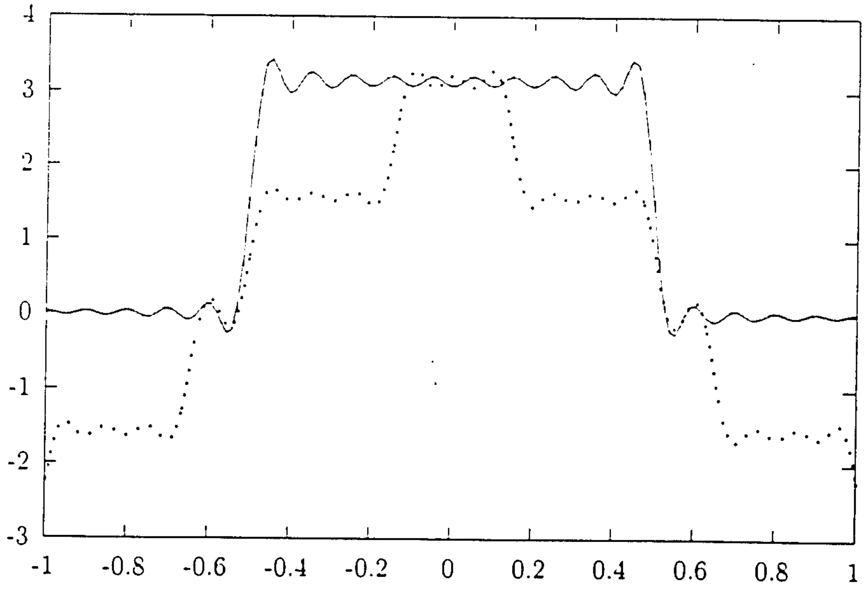


Figure 5c: The optimal approximation to the sector pattern generated by 24 equispaced nodes, as described in Example 5.1

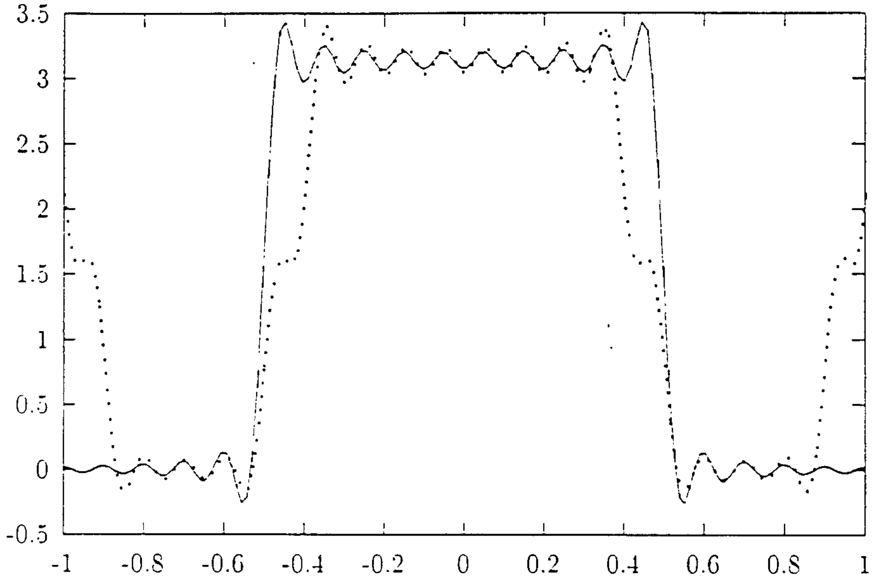


Figure 5d: The optimal approximation to the sector pattern generated by 29 equispaced nodes, as described in Example 5.1

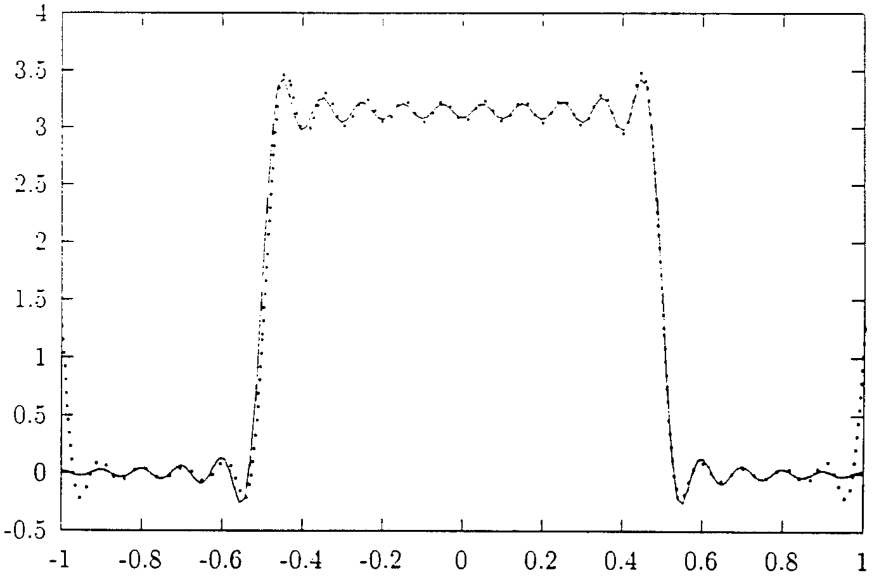


Figure 5e: The optimal approximation to the sector pattern generated by 31 equispaced nodes, as described in Example 5.1

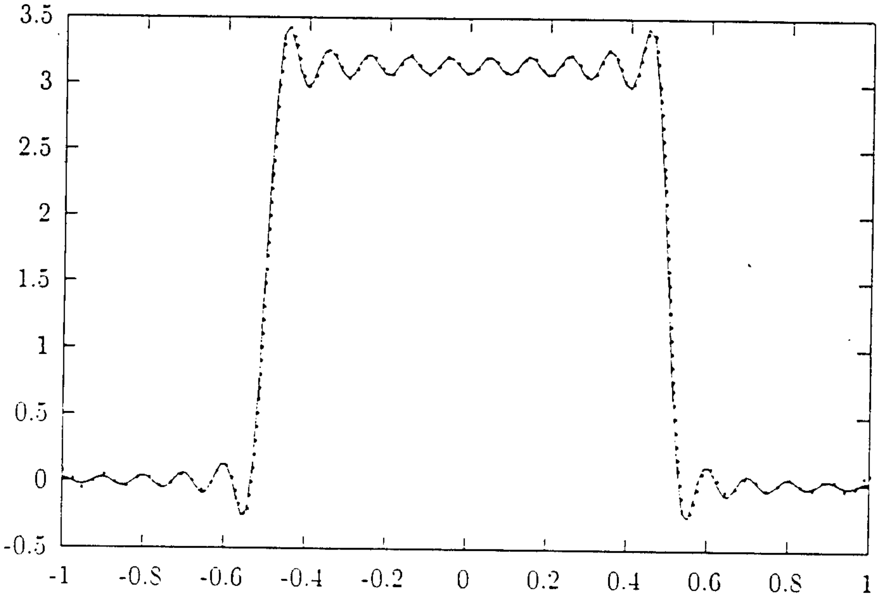


Figure 5f: The optimal approximation to the sector pattern generated by 34 equispaced nodes, as described in Example 5.1

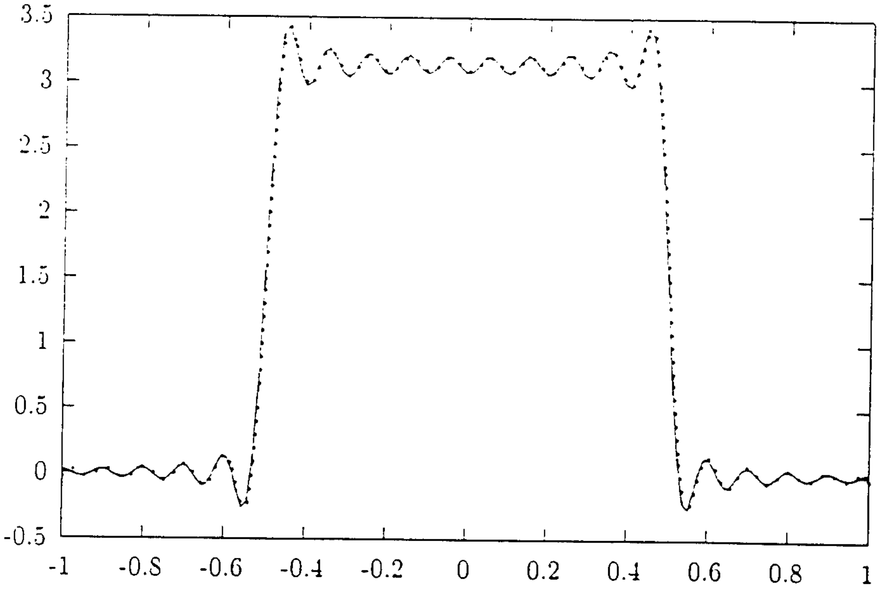


Figure 5g: The optimal approximation to the sector pattern generated by 21 optimal nodes, as described in Example 5.1

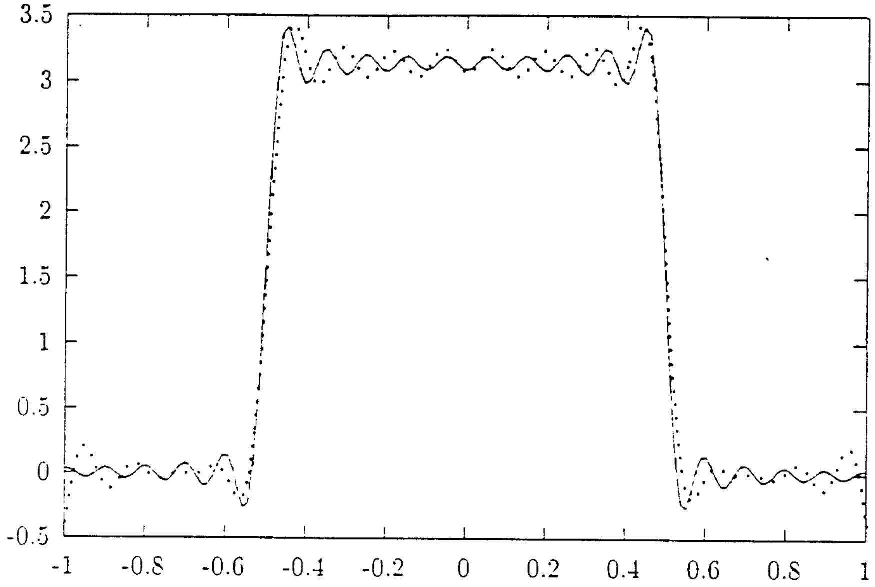


Figure 5h: The optimal approximation to the sector pattern generated by 17 optimal nodes, as described in Example 5.1

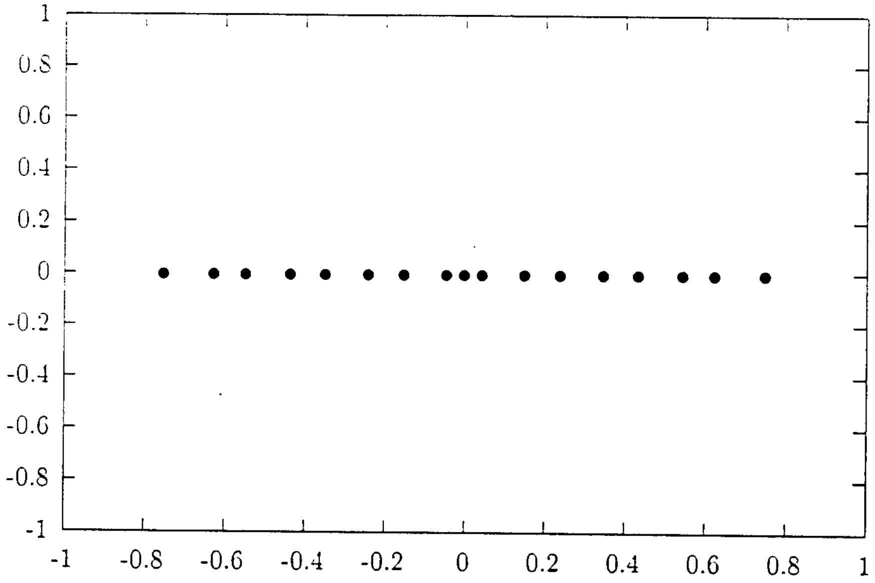


Figure 5i: The distribution of 17 elements creating the pattern depicted in Figure 5h, as described in Example 5.1

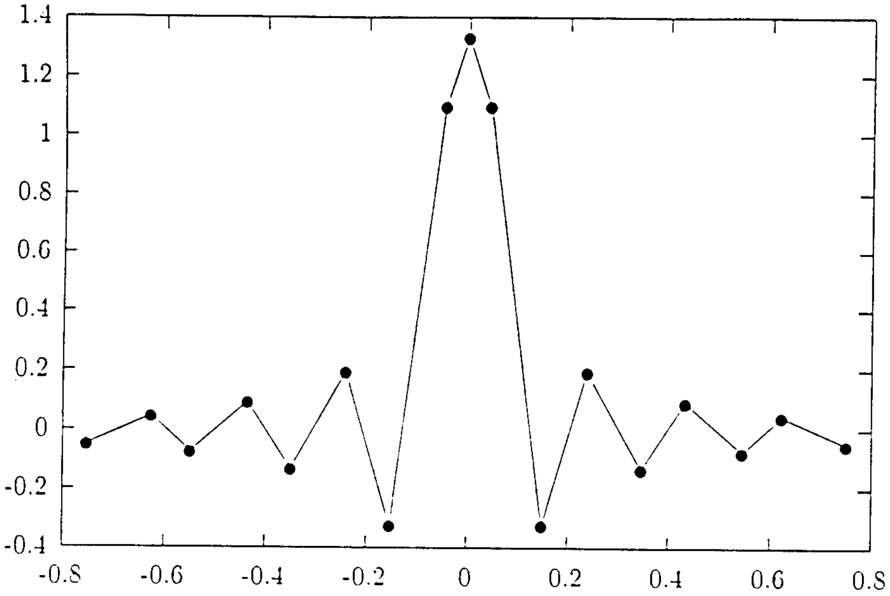


Figure 5j: The values of the sources located at the nodes depicted in Figure 5i and generating the pattern depicted in Figure 5h. as described in Example 5.1

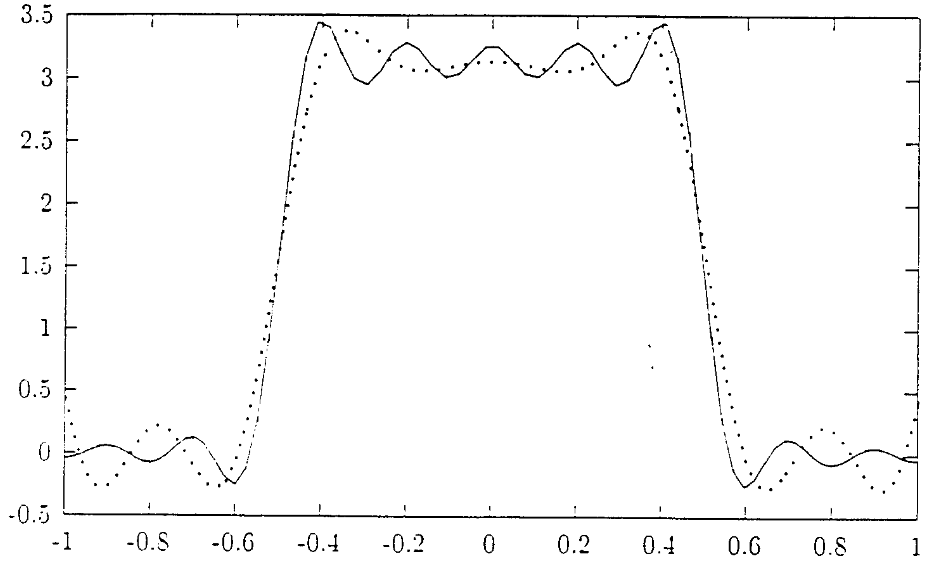


Figure 6: The pattern created by the 9 optimal elements, depicted in Figure 6a as described in Example 5.2

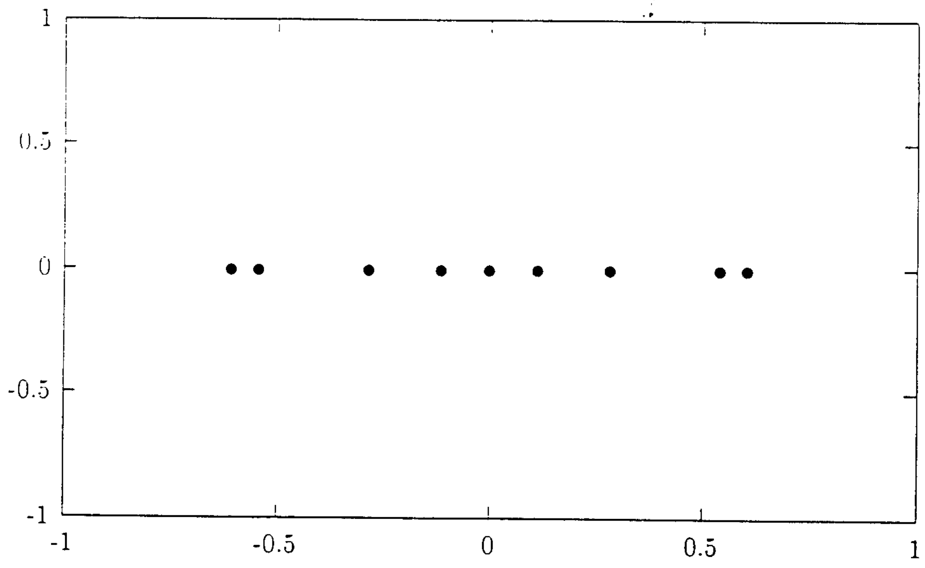


Figure 6a: The distribution of elements creating the pattern depicted in Figure 6, as described in Example 5.2

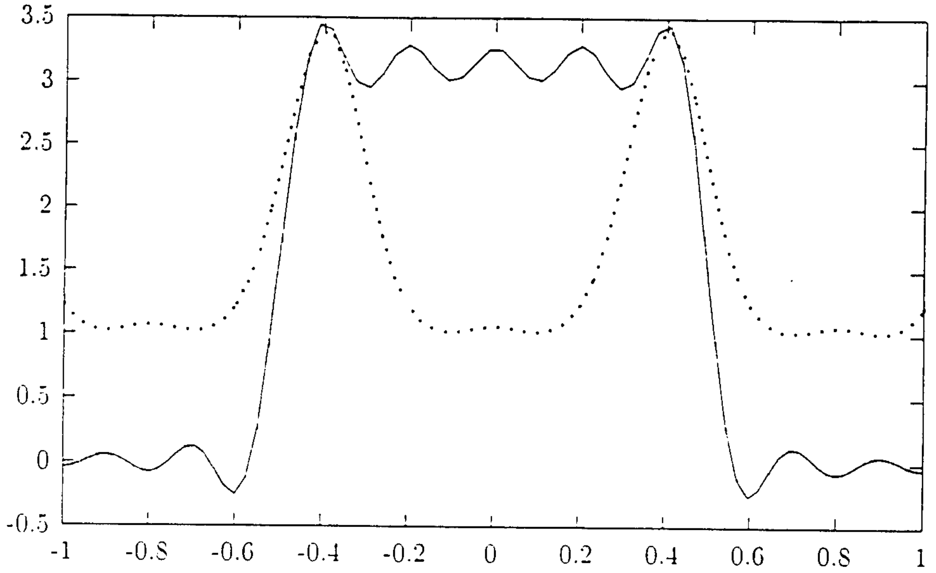


Figure 6b: The optimal approximation to the sector pattern generated by 9 equispaced nodes, as described in Example 5.2

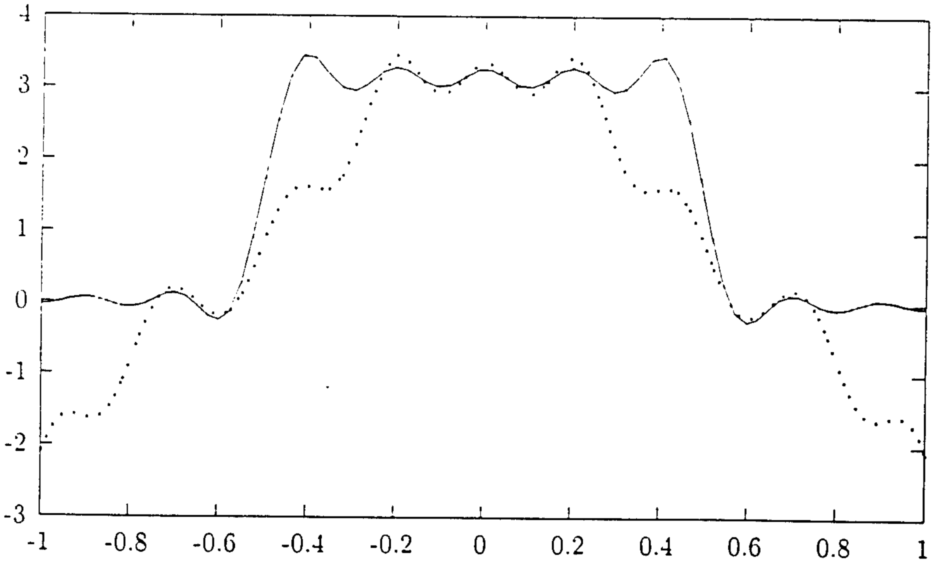


Figure 6c: The optimal approximation to the sector pattern generated by 14 equispaced nodes, as described in Example 5.2

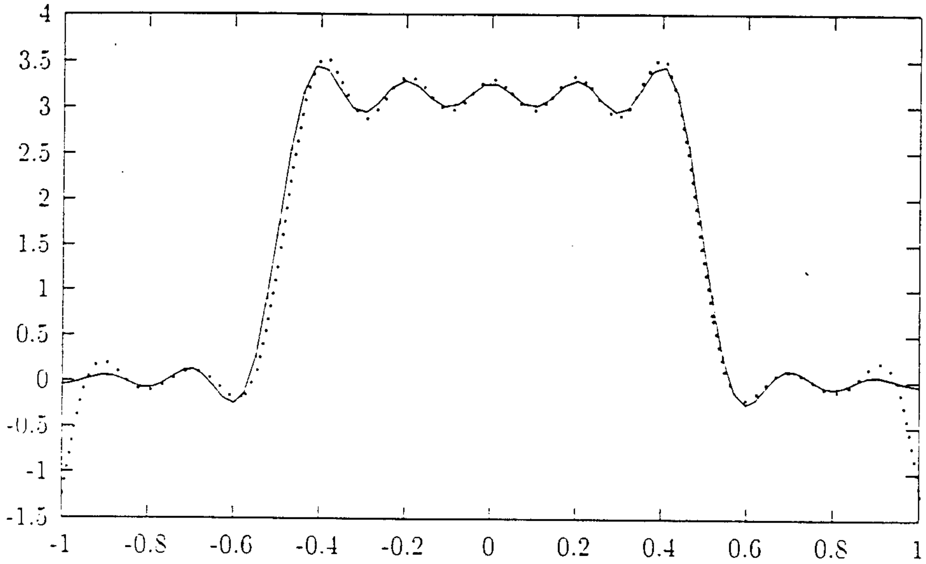


Figure 6d: The optimal approximation to the sector pattern generated by 16 equispaced nodes, as described in Example 5.2

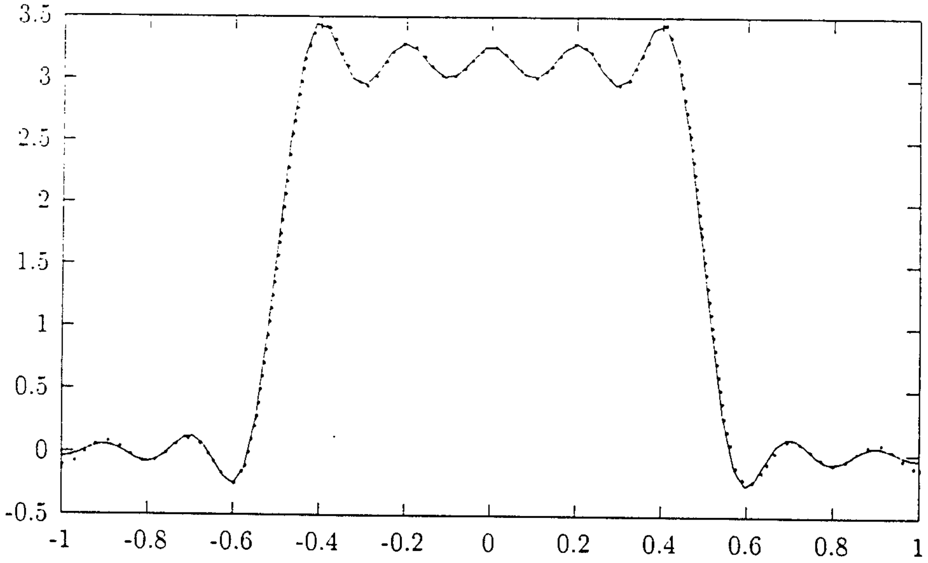


Figure 6e: The optimal approximation to the sector pattern generated by 18 equispaced nodes, as described in Example 5.2

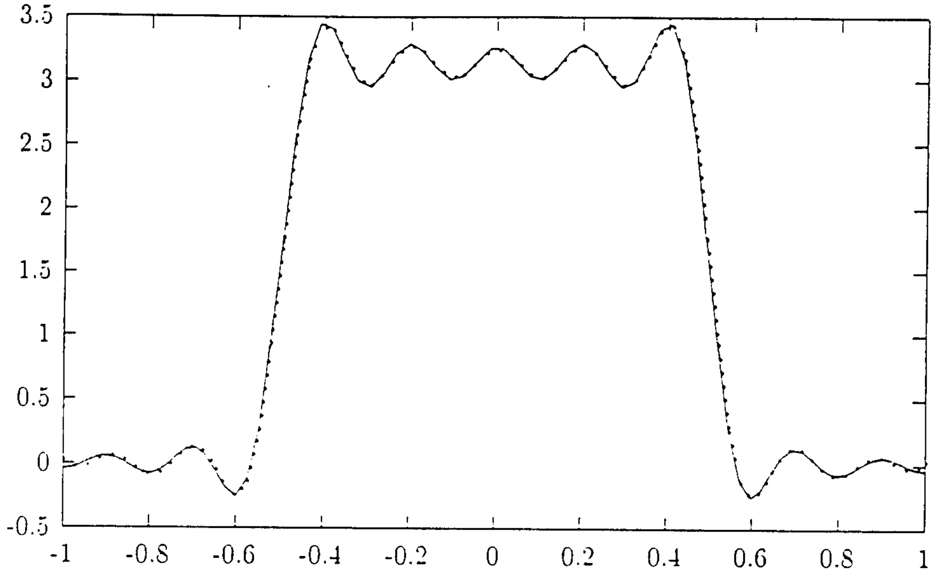


Figure 6f: The pattern created by the 11 optimal elements. in Example 5.2

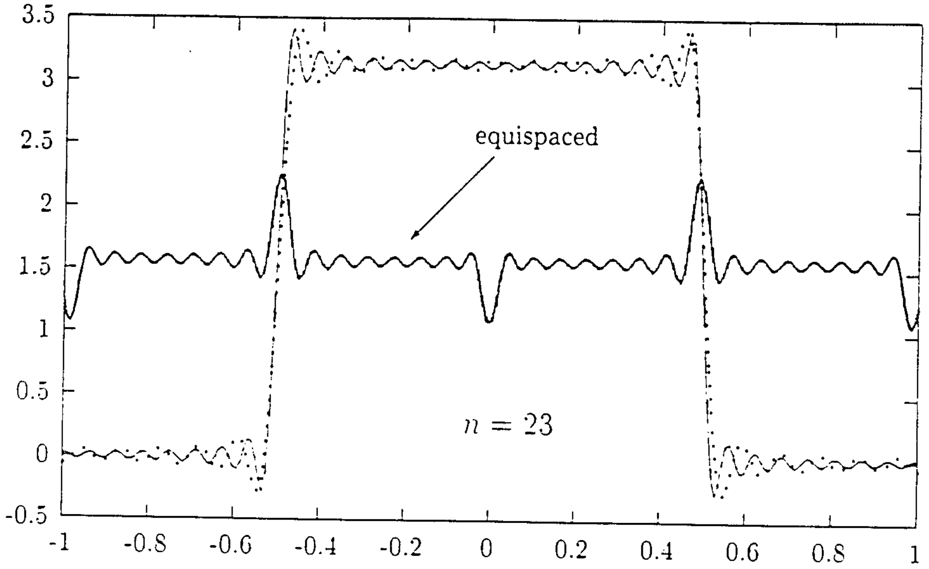


Figure 7a: The approximation to the sector pattern generated by 23 optimal elements, vs. optimal approximation by 23 equispaced nodes, as described in Example 5.3

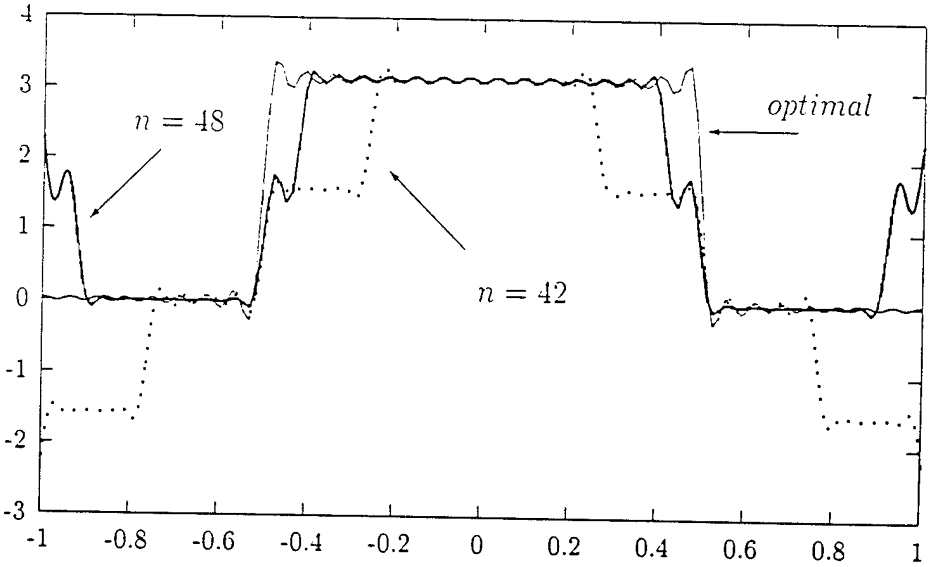


Figure 7b: The optimal approximations to the sector pattern generated by 42 and 48 equispaced nodes, as described in Example 5.3

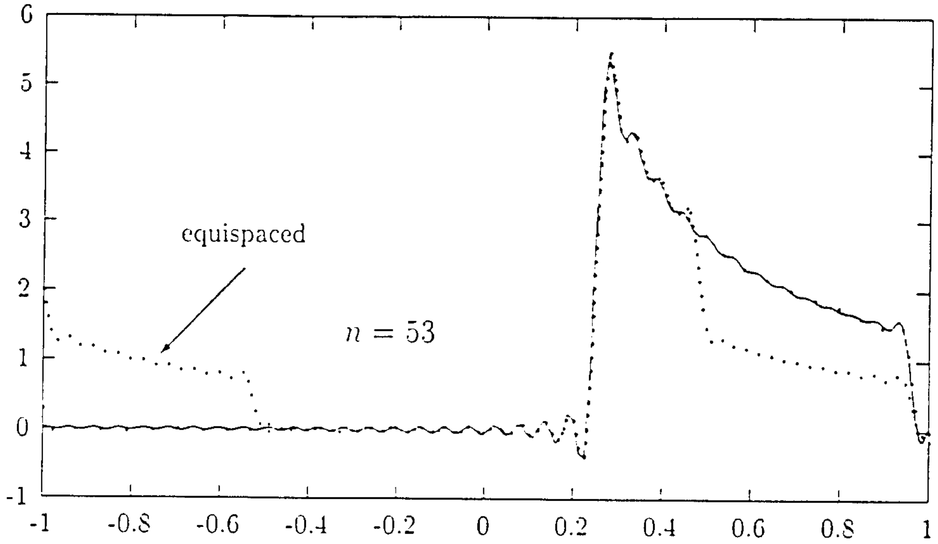


Figure 8a: The approximation to the cosecant pattern generated by 53 optimal elements, vs. optimal approximation by 53 equispaced nodes, as described in Example 5.4

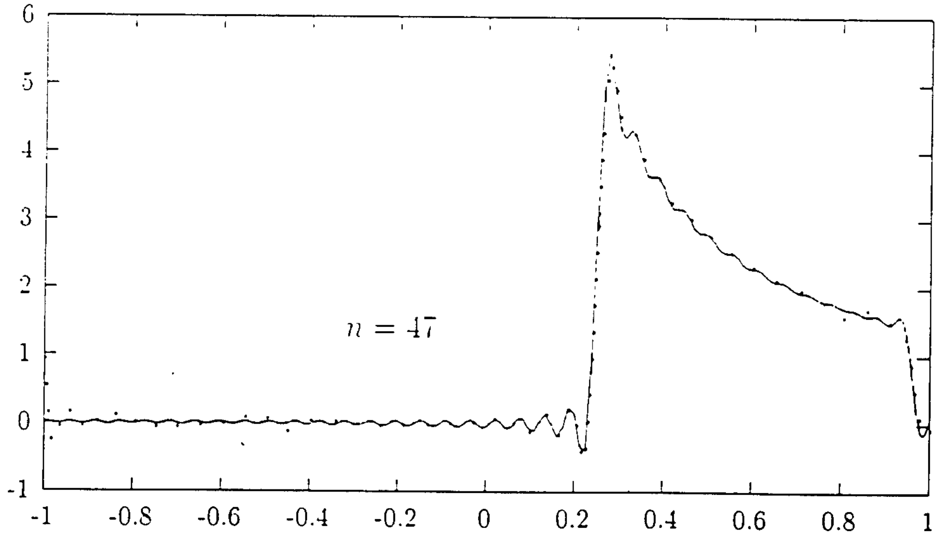


Figure 8a: The approximation to the cosecant pattern generated by 47 optimal elements, as described in Example 5.4

References

[1] M. Abramovitz, I. Stegun, *Handbook of Mathematical Functions*, Applied Math. Series (National Bureau of Standards), Washington, DC, 1964.

[2] H. Cheng, N. Yarvin, V. Rokhlin, *Non-Linear Optimization, Quadrature, and Interpolation*, Yale University Technical Report, YALEU/DCS/RR-1169, 1998, to appear in the SIAM Journal of Non-linear Optimization.

[3] F. GANTMACHER AND M. KREIN, *Oscillation matrices and kernels and small oscillations of mechanical systems*, 2nd ed., Gosudarstv. Izdat. Tehn-Teor. Lit., Moscow, 1950 (Russian).

[4] F. A. Grünbaum, *Toeplitz Matrices Commuting With Tridiagonal Matrices*, J. Linear Alg. and Appl., 40, (1981).

[5] F. A. Grünbaum, *Eigenvectors of a Toeplitz Matrix: Discrete Version of the Prolate Spheroidal Wave Functions*, SIAM J. Alg. Disc. Math., 2(1981).

[6] F. A. Grünbaum, L. Longhi, M. Perlstadt, *Differential Operators Commuting with Finite Convolution Integral Operators: Some Non-Abelian Examples*, SIAM J. Appl. Math. 42(1982).

[7] S. KARLIN, *The Existence of Eigenvalues for Integral Operators*, Trans. Am. Math. Soc. v. 113, pp. 1-17 (1964).

[8] S. KARLIN, AND W. J. STUDDEN, *Tchebycheff Systems with Applications In Analysis And Statistics*, John Wiley (Interscience), New York, 1966.

[9] John D. Kraus, *Antennas*, McGraw-Hill, 1988.

[10] M. G. KREIN, *The Ideas of P. L. Chebyshev and A. A. Markov in the Theory Of Limiting Values Of Integrals*, American Mathematical Society Translations, Ser. 2, Vol. 12, 1959, pp. 1-122.

[11] H.J. Landau, H. Widom, *Eigenvalue Distribution of Time and Frequency Limiting*, Journal of Mathematical Analysis and Applications, 77, 469-481 (1980).

[12] Y.T. Lo, S.W. Lee, editors, *Antenna Handbook, Theory, Applications, and Design*, Van Nostrand Reinhold Company, 1988.

[13] J. MA, V. ROKHLIN, AND S. WANDZURA, *Generalized Gaussian Quadratures For Systems of Arbitrary Functions*, SIAM Journal of Numerical Analysis, v. 33, No. 3, pp. 971-996, 1996.

[14] R.J. Mailloux, *Phased Array Antenna Handbook*, Artech House, 1994.

[15] A. A. MARKOV, *On the limiting values of integrals in connection with interpolation*, Zap. Imp. Akad. Nauk. Fiz.-Mat. Otd. (8) 6 (1898), no.5 (Russian), pp. 146-230 of [16].

[16] A. A. MARKOV, *Selected papers on continued fractions and the theory of functions deviating least from zero*, OGIZ, Moscow-Leningrad, 1948 (Russian).

[17] P.M. Morse, H. Feshbach, *Methods of Theoretical Physics*, McGraw-Hill, New York, 1953.

[18] D. Rhodes, *The optimum line source for the best mean-square approximation to a given radiation pattern*, IEEE Trans. AP, July 1963.

[19] D. Rhodes, *Synthesis of planar antenna sources*, Clarendon Press, Oxford, 1974.

[20] D. Slepian, H.O. Pollak, *Prolate Spheroidal Wave Functions, Fourier Analysis, and Uncertainty - I*, The Bell System Technical Journal, January 1961.

[21] H.J. Landau, H.O. Pollak, *Prolate Spheroidal Wave Functions, Fourier Analysis, and Uncertainty - II*, The Bell System Technical Journal, January 1961.

[22] H.J. Landau, H.O. Pollak, *Prolate Spheroidal Wave Functions, Fourier Analysis, and Uncertainty - III: The Dimension of Space of Essentially Time- and Band-Limited Signals*, The Bell System Technical Journal, July 1962.

[23] D. Slepian, *Prolate Spheroidal Wave Functions, Fourier Analysis, and Uncertainty - IV: Extensions to Many Dimensions, Generalized Prolate Spheroidal Wave Functions*, The Bell System Technical Journal, November 1964.

[24] D. Slepian, *Prolate Spheroidal Wave Functions, Fourier Analysis, and Uncertainty - V: The Discrete Case*, The Bell System Technical Journal, May-June 1978.

[25] D. Slepian, *Some Comments on Fourier Analysis, Uncertainty, and Modeling* SIAM Review, V. 25, No. 3, July 1983.

[26] W.L. Stutzman, G.A. Thiele, *Antenna Theory and Design*, Wiley, 1998.

[27] T.T. Taylor, *Design of Line-Source Antennas for Narrow Beamwidth and Low Side Lobes*, IEEE Trans. on Antennas and Propagation, AP-3, pp. 16-28, 1955.

[28] N. Yarvin and V. Rokhlin, *Generalized Gaussian Quadratures and Singular Value Decompositions of Integral Operators*, SIAM Journal of Scientific Computing, Vol. 20, No. 2, pp. 699-718 (1998).

A Generalized Fast Multipole Method for Non-Oscillatory Kernels

Z. Gimbutas* and V. Rokhlin†
Research Report YALEU/DCS/RR-1202
July 31, 2000

YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

We present a modification of the Fast Multipole Method (FMM) in two dimensions. While previous implementations of the FMM have been designed for harmonic kernels, our algorithm works for a large class of kernels that satisfy fairly general conditions, amounting to the kernel being sufficiently smooth away from the diagonal. Our algorithm approximates appropriately chosen parts of the kernel with "tensor products" of Legendre expansions and uses the Singular Value Decomposition (SVD) to compress the resulting representations. The obtained singular function expansions replace the Taylor and Laurent expansions used in the original FMM. The algorithm requires $O(N)$ operations, and is stable and robust. The performance of the algorithm is illustrated with numerical examples.

A Generalized Fast Multipole Method for Non-Oscillatory Kernels

Z. Gimbutas* and V. Rokhlin†

Research Report YALEU/DCS/RR-1202
July 31, 2000

---

* The author was supported in part by AFOSR under Grant F49620/97/1/0011, and in part by ONR under Grant N00014/89/J/1527.

† The author was supported in part by DARPA/AFOSR under Contract F49620/97/1/0011, and in part by AFOSR under STTR number F49620/98/C/0051

Approved for public release: distribution is unlimited

Keywords: *Fast Multipole Method, Arbitrary Kernels, Potential Theory.*

1 Introduction

In this paper, we describe a fast algorithm for the evaluation of all pairwise interactions in large ensembles of particles in the plane, i.e., sums of the form

$$u(x_i) = \sum_{j=1}^{N} q_j K(x_i, x_j), \tag{1}$$

where $q_1, \ldots, q_N$ are arbitrary complex numbers, $x_1, \ldots, x_n$ are points in the plane, and $K : R^2 \to R^2$ is a non-oscillatory kernel. Such computations appear in a variety of numerical methods for the solution of problems of computational physics.

The algorithm of this paper is a version of the Fast Multipole Method (FMM) in two dimensions. The structure of the FMM algorithm is left virtually unchanged from the one described by in [3]. The version of the FMM algorithm used in this paper, however, replaces the Taylor and Laurent expansions with "tensor products" of Legendre expansions that are subsequently compressed via the Singular Value Decomposition (SVD). This approach leads to an algorithm that can be applied to a variety of non-oscillatory kernels that are sufficiently smooth away from the diagonal.

In two dimensions, the original Fast Multipole Method (FMM) relies on the Taylor and Laurent expansions (see [14], [7]) for the evaluation of Coulomb interactions in large ensembles of particles. During the last decade, several improvements of the original scheme have been suggested. A new version of the FMM, based on specially designed singular function expansions, was introduced in [10]. The approach taken in the latter paper, when used in combination with an intermediate representation consisting of complex exponentials, leads to an algorithm that is about five times as fast as the original FMM, due to the reduction of the number of parameters needed to represent far and near fields. A similar technique was used in one dimension in [18]. A version of the FMM for polynomial interpolation (see [5]) uses Chebyshev expansions that are compressed by a suitable change of basis obtained via Singular Value Decomposition (SVD). Finally, an analytical apparatus based on least squares approximation of integral operators was developed in [17]. This analytical apparatus leads to fast algorithms for a fairly large class of kernels in one dimension.

The plan of the paper is as follows. In Section 2, we introduce mathematical and numerical preliminaries. In Sections 3 and 4, we describe a generalized Fast Multipole Method in two dimensions and present the complexity analysis. Finally, in Section 5, we demonstrate the performance of the algorithm with several numerical examples.

2 Mathematical Preliminaries

2.1 Gaussian Integration and Interpolation

In what follows, we will denote by $P_n^{a,b}$ the n-th Legendre polynomial on the interval $[a, b] \subset R$. We will refer to the roots $x_1^{a,b}, \ldots, x_n^{a,b}$ of $P_n^{a,b}(x)$ as the Gaussian nodes of order n and will denote by $w_1^{a,b}, \ldots, w_n^{a,b}$ the weights of the corresponding Gaussian quadrature on the interval $[a, b]$. We will denote by L_n the projection from the space of continuous functions on the interval $[a, b]$ to the space of polynomials of order n, preserving the function values at the Gaussian nodes. For a given continuous function $f : [a,b] \to C$, the function $L_n f(x)$ is the polynomial of order n such that $L_n f(x_m^{a,b}) = f(x_m^{a,b})$. As is well known, for all $x \in [a,b]$,

$$L_n f(x) = \sum_{k=0}^{n-1} a_k \cdot P_k^{a,b}(x), \tag{2}$$

and the coefficients a_k are given by the formula

$$a_k = \sum_{m=1}^{n} w_m^{a,b} \cdot f(x_m^{a,b}) \cdot P_k(x_m^{a,b}). \tag{3}$$

The polynomial $L_n f$ will be referred to as n-th order Legendre expansion of the function f. For any integer n we will denote by $||L_n||_\infty$ the L^∞-norm of the operator L_n, defined by the formula

$$||L_n||_\infty = \sup_{||f||_{L^\infty[a,b]}=1} ||L_n f||_{L^\infty[a,b]}. \tag{4}$$

We will denote by $\alpha_1(x), \dots, \alpha_n(x)$ the set of polynomials of order n defined by the formulae

$$\alpha_i(x) = \prod_{k=1, k\neq i}^{n} \frac{x - x_k}{x_i - x_k}, \quad i = 1, 2, \dots, n, \tag{5}$$

where $x_1, \dots, x_k$ are the Gaussian nodes of order n on the interval $[a,b]$. It is readily seen from (5) that for any continuous function $f : [a,b] \to C$,

$$L_n f(x) = \sum_{k=0}^{n-1} a_k \cdot P_k(x) = \sum_{i=1}^{n} f(x_i) \cdot \alpha_i(x). \tag{6}$$

For any natural n and continuous function $f : [a,b] \to C$, we will denote by $E_n f$ the error of the best approximation to f among all polynomials of order n, i.e.,

$$E_n f = \min_P ||f - P||_{L^\infty[a,b]}. \tag{7}$$

Let $\rho > 0$ be an arbitrary positive real number. For any analytic function $f : C \to C$, we will denote by $M([a,b], f, \rho)$ the maximum of the absolute value of f in the ρ-neighborhood of the interval $[a,b]$, i.e.,

$$M([a,b], f, \rho) = \sup_{x\in[a,b]} \sup_{\theta\in[-\pi,\pi]} |f(x + \rho e^{i\theta})|. \tag{8}$$

The following five lemmas are well known. Their proofs can be found, for example, in [16], [12].

Lemma 2.1. *If $n > 0$ is an integer, and $P : C \to C$ is a polynomial of order n, then for any interval $[a,b] \subset R$,*

$$\frac{1}{\sqrt{|b-a|}} ||P||_{L^2[a,b]} \le ||P||_{L^\infty[a,b]} \le \frac{n}{\sqrt{|b-a|}} ||P||_{L^2[a,b]}. \tag{9}$$

Lemma 2.2. *For any continuous function* $f : [a, b] \to C$,

$$||f - L_n f||_{L^\infty[a,b]} \leq (1 + ||L_n||_\infty) \cdot ||f - E_n f||_{L^\infty[a,b]}. \tag{10}$$

Lemma 2.3. *For any* n *times continuously differentiable function* $f : [a, b] \to C$,

$$||f - E_n f||_{L^\infty[a,b]} \leq \frac{2(b-a)^n}{4^n n!} \cdot ||f^{(n)}||_{L^\infty[a,b]}. \tag{11}$$

Lemma 2.4. *If* $f : C \to C$ *is an analytic function, then for any positive real* $\rho > 0$,

$$||f^{(n)}||_{L^\infty[a,b]} \leq n! \cdot \frac{M([a,b], f, \rho)}{\rho^n}. \tag{12}$$

Lemma 2.5. *For any natural* n,

$$||L_n||_\infty \leq n. \tag{13}$$

By combining (9), (10), (11), (12), and (13), we obtain the following theorem describing the rate of convergence of Legendre expansions of an analytic function on the interval $[a, b]$.

Lemma 2.6. *Suppose that* $f : C \to C$ *is an analytic function, and that for some positive* $\rho > (b - a)/4$,

$$M([a,b], f, \rho) < \infty. \tag{14}$$

Then

$$\lim_{n\to\infty} ||f - L_n f||_{L^\infty[a,b]} = 0. \tag{15}$$

Furthermore, for any $n \geq 1$,

$$||f - L_n f||_{L^\infty[a,b]} \leq 2(1 + n) \cdot M([a,b], f, \rho) \cdot \left(\frac{b-a}{4\rho}\right)^n. \tag{16}$$

A standard approach to the construction of polynomial approximations of functions in higher dimensions is to expand them into "tensor products" of one-dimensional Legendre polynomials. For an m-dimensional cube $Q = [a_1, b_1] \times \ldots \times [a_m, b_m]$ and continuous function $f : Q \to C$, we will denote by $L_n f$ the (unique) polynomial of m variables having the form

$$L_n f(x_1, \ldots, x_m) = \sum_{k_1=0}^{n-1} \cdots \sum_{k_m=0}^{n-1} a_{k_1,\ldots,k_m} \cdot P_{k_1}^{a_1,b_1}(x_1) \cdot \ldots \cdot P_{k_m}^{a_m,b_m}(x_m), \tag{17}$$

and coinciding with f on the n^m "tensor product" Gaussian nodes

$$(x_{k_1}^{a_1,b_1}, \ldots, x_{k_m}^{a_m,b_m}), \quad k_1 = 1, \ldots, n; \ldots; k_m = 1, \ldots, n; \tag{18}$$

the coefficients $a_{k_1,\ldots,k_m}$ are given by the formula

$$a_{k_1,\ldots,k_m} = \sum_{k_1=0}^{n-1} \cdots \sum_{k_m=0}^{n-1} w_{k_1}^{a_1,b_1} \cdot \ldots \cdot w_{k_m}^{a_m,b_m} \cdot f(x_{k_1}^{a_1,b_1}, \ldots, x_{k_m}^{a_m,b_m}) \cdot P_{k_1}^{a_1,b_1}(x_{k_1}^{a_1,b_1}) \cdot \ldots \cdot P_{k_m}^{a_m,b_m}(x_{k_m}^{a_m,b_m}). \tag{19}$$

In a mild abuse of terminology, we will be referring to such polynomials as polynomials of order n in R^m and to expansions of the form (17) as Legendre expansions of order n in the cube $Q \in R^m$. For an analytic function $f : C^m \to C$, we will denote by $M(Q, f, \rho)$ the maximum of the absolute value of f in the ρ-neighborhood of the cube Q, i.e.,

$$M(Q,f,\rho) = \max_{k=1,\ldots,m} \sup_{x\in Q} \sup_{\theta\in[-\pi,\pi]} |f(x_1,\ldots,x_k+\rho e^{i\theta},\ldots,x_m)|. \tag{20}$$

The following two lemmas are a simple consequence of Lemmas 2.1 and 2.6; they can be viewed as multidimensional analogues of the latter (see for example [17]).

Lemma 2.7. *If $n > 0$ is an integer and $P : C^m \to C$ is a polynomial of order n, then for any cube $Q = [a,b]^m \subset R^m$,*

$$\frac{1}{|b-a|^{m/2}}||P||_{L^2(Q)} \le ||P||_{L^\infty(Q)} \le \frac{n^m}{|b-a|^{m/2}}||P||_{L^2(Q)}. \tag{21}$$

Lemma 2.8. *Suppose that $f : C^m \to C$ is an analytic function on C^m, and that for some positive $\rho > (b-a)/4$,*

$$M([a,b]^m, f, \rho) < \infty. \tag{22}$$

Then, for any $n \ge 1$,

$$||f - L_n f||_{L^\infty[a,b]^m} \le 2(1+n)^m \cdot M([a,b]^m, f, \rho) \cdot \left(\frac{b-a}{4\rho}\right)^n. \tag{23}$$

2.2 Singular Value Decomposition of Integral Operators

Let $T : L^2(Y) \to L^2(X)$ be integral operator given by the formula

$$(T\cdot f)(x) = \int_Y K(x,y) f(y)\, dy, \tag{24}$$

where K is a square integrable function on $X \times Y$, i.e.,

$$||K(x,y)||_{L^2(X\times Y)} = \left(\int\int_{X\times Y} |K(x,y)|^2\, dx\, dy\right)^{1/2} < +\infty. \tag{25}$$

The function $K : X \times Y \to R$ is usually referred to as the kernel of the integral operator T.

The following theorem can be found (in a more general form) in [15].

Theorem 2.9. *For any $K \in L^2(X \times Y)$, there exist two orthonormal systems of functions $\{u_k\} \in L^2(X)$, $\{v_k\} \in L^2(Y)$, and a sequence of nonnegative numbers $s_1 \ge s_2 \ge \ldots \ge 0$, for $k = 1, 2, \ldots$, such that*

$$K(x,y) = \sum_{k=1}^{\infty} u_k(x) s_k v_k(y), \tag{26}$$

in $L^2(X \times Y)$ sense,

$$\sum_{k=1}^{\infty} |s_k|^2 < +\infty, \tag{27}$$

and the sequence $\{s_k\}$ is uniquely determined by K.

Formula (26) is normally referred to as the singular value decomposition (SVD) of the operator T (or the kernel K). The functions u_k and v_k are usually referred to as the left and the right singular functions, respectively, and the numbers s_k are referred to as singular values of the operator K (or the kernel K).

The singular value decomposition can be used to construct finite-dimensional approximations to the operators of the form (24) and the corresponding kernels K. Specifically, given a positive real $\varepsilon > 0$, one can truncate the expression (26) after a finite number p of terms, leading to the expression

$$K(x,y) \approx \sum_{k=1}^{p} u_k(x) s_k v_k(y). \tag{28}$$

Now, if p has been chosen in such a manner that

$$\sqrt{\sum_{k=p+1}^{\infty} s_k^2} \leq \varepsilon, \tag{29}$$

then due to (26),

$$||K(x,y) - \sum_{k=1}^{p} u_k(x) s_k v_k(y)||_{L^2(X\times Y)} \leq \varepsilon. \tag{30}$$

Theorem 2.10 (Minimal property of the SVD). *Suppose that the SVD of the operator* $T : L^2(Y) \to L^2(X)$ *with the kernel* $K : X \times Y \to R$ *is given by the formula*

$$K(x,y) = \sum_{k=1}^{\infty} u_k(x) s_k v_k(y). \tag{31}$$

Then for any $f \in L^2(Y)$,

$$||(T \cdot f)(x) - \sum_{k=1}^{p} u_k(x) s_k b_k||_{L^2(X)} \leq s_{p+1} ||f||_{L^2(Y)}, \tag{32}$$

where the coefficients b_k *are given by the formula*

$$b_k = \int_Y f(y) v_k(y)\, dy. \tag{33}$$

2.3 Approximation of the SVD of Integrals Operators

The following theorem is a straightforward generalization of Theorem 2.10.

Theorem 2.11 (Approximation of the SVD). *Suppose that the operator* $T : L^2(Y) \to L^2(X)$ *is defined by (24), that there exist a positive number* $\delta > 0$ *and a square integrable function* $\tilde{K} : X \times Y \to R$ *such that*

$$||K(x,y) - \tilde{K}(x,y)||_{L^2(X\times Y)} \leq \delta, \tag{34}$$

and that the SVD of $\tilde{K}$ is given by the formula

$$\tilde{K}(x,y) = \sum_{k=1}^{\infty} \tilde{u}_k(x)\tilde{s}_k\tilde{v}_k(y). \tag{35}$$

Then for any $f \in L^2(Y)$,

$$||(T \cdot f)(x) - \sum_{k=1}^{p} \tilde{u}_k(x)\tilde{s}_k\tilde{b}_k||_{L^2(X)} \le (\delta + \tilde{s}_{p+1})||f||_{L^2(Y)}, \tag{36}$$

where the coefficients $\tilde{b}_k$ are given by the formula

$$\tilde{b}_k = \int_Y f(y)\tilde{v}_k(y)\,dy. \tag{37}$$

Proof. Obviously, (34) implies

$$||\int_Y K(x,y)f(y)\,dy - \int_Y \tilde{K}(x,y)f(y)\,dy||_{L^2(X)} \le \delta||f||_{L^2(Y)}, \tag{38}$$

and from Theorem 2.10, we obtain

$$||\int_Y \tilde{K}(x,y)f(y)\,dy - \sum_{k=1}^{p} \tilde{u}_k(x)\tilde{s}_k\tilde{b}_k||_{L^2(X)} \le \tilde{s}_{p+1}||f||_{L^2(Y)}. \tag{39}$$

Now, (36) follows immediately from (38), (39), and the triangle inequality. □

3 Analytical Apparatus

In the remainder of this paper, we will be assuming that all charges are located in a unit square $[0,1] \times [0,1]$ in $\mathbf{R}^2$.

3.1 Notation

We will denote by $Y^{(l,k_1,k_2)}$ the square

$$Y^{(l,k_1,k_2)} = \left[\frac{k_1-1}{2^l}, \frac{k_1}{2^l}\right] \times \left[\frac{k_2-1}{2^l}, \frac{k_2}{2^l}\right], \tag{40}$$

where $l \ge 1$, $k_1 = 1, \ldots, 2^l$, $k_2 = 1, \ldots, 2^l$; l will be referred to as the level of the square $Y^{(l,k_1,k_2)}$, and (k_1, k_2) will be referred to as the coordinates of the square $Y^{(l,k_1,k_2)}$. We will denote by $Z^{(l,k_1,k_2)}$ the union of the square $Y^{(l,k_1,k_2)}$ and its immediate neighbors on the level l. We will denote the subset $X^{(l,k_1,k_2)}$ of $[0,1] \times [0,1]$ by the formula

$$X^{(l,k_1,k_2)} = [0,1] \times [0,1] \setminus Z^{(l,k_1,k_2)}, \tag{41}$$

and refer to $X^{(l,k_1,k_2)}$ as the interaction domain of the square $Y^{(l,k_1,k_2)}$. In other words, the interaction domain of the square $Y^{(l,k_1,k_2)}$ consists of all squares on level l that are not immediate neighbors of $Y^{(l,k_1,k_2)}$ and not $Y^{(l,k_1,k_2)}$ itself. For consistency, we will also referring to the unit square $[0,1]\times[0,1]$ as $Y^{(0,1,1)}$.

Suppose now that the function $K: Y^{(0,1,1)} \times Y^{(0,1,1)} \to C$ is such that

$$\int_{X^{(l,k_1,k_2)}} \left(\int_{Y^{(l,k_1,k_2)}} |K(x,y)|^2 \, dy \right) dx < +\infty, \tag{42}$$

and

$$\int_{Y^{(l,k_1,k_2)}} \left(\int_{X^{(l,k_1,k_2)}} |K(y,x)|^2 \, dx \right) dy < +\infty, \tag{43}$$

for all $l \geq 1$, $k_1 = 1, \ldots, 2^l$, $k_2 = 1, \ldots, 2^l$. For any square $Y^{(l,k_1,k_2)}$, we will define the integral operators

$$\begin{aligned} P^{(l,k_1,k_2)} &: L^2(Y^{(l,k_1,k_2)}) \to L^2(X^{(l,k_1,k_2)}), &(44)\\ R^{(l,k_1,k_2)} &: L^2(X^{(l,k_1,k_2)}) \to L^2(Y^{(l,k_1,k_2)}), &(45)\end{aligned}$$

by the formulae

$$\begin{aligned} (P^{(l,k_1,k_2)} \cdot \sigma)(x) &= \int_{Y^{(l,k_1,k_2)}} K(x,y)\sigma(y)\, dy, &(46)\\ (R^{(l,k_1,k_2)} \cdot \sigma)(y) &= \int_{X^{(l,k_1,k_2)}} K(y,x)\sigma(x)\, dx. &(47)\end{aligned}$$

The function $(P^{(l,k_1,k_2)} \cdot \sigma) \in L^2(X^{(l,k_1,k_2)})$ with $\sigma \in L^2(Y^{(l,k_1,k_2)})$ will be referred to as the potential due to the charge distribution σ on the square $Y^{(l,k_1,k_2)}$. Similarly, the function $(R_{r_1}^{(l,k_1,k_2)} \cdot \sigma) \in L^2(Y^{(l,k_1,k_2)})$ with $\sigma \in L^2(X^{(l,k_1,k_2)})$ will be referred to as the incoming potential due to some charge distribution σ on $X^{(l,k_1,k_2)}$.

Due to (42), (43), and Theorem 2.9, there exist functions

$$\{u_k^{in,(l,k_1,k_2)}\} \in L^2(Y^{(l,k_1,k_2)}), \quad \{v_k^{out,(l,k_1,k_2)}\} \in L^2(Y^{(l,k_1,k_2)}), \tag{48}$$

$$\{u_k^{out,(l,k_1,k_2)}\} \in L^2(X^{(l,k_1,k_2)}), \quad \{v_k^{in,(l,k_1,k_2)}\} \in L^2(X^{(l,k_1,k_2)}), \tag{49}$$

and positive real numbers

$$\{s_k^{in,(l,k_1,k_2)}\}, \quad \{s_k^{out,(l,k_1,k_2)}\}, \tag{50}$$

such that

$$\begin{aligned} K(x,y) &= \sum_{k=1}^{\infty} u_k^{out,(l,k_1,k_2)}(x) s_k^{out,(l,k_1,k_2)} v_k^{out,(l,k_1,k_2)}(y), &(51)\\ K(y,x) &= \sum_{k=1}^{\infty} u_k^{in,(l,k_1,k_2)}(y) s_k^{in,(l,k_1,k_2)} v_k^{in,(l,k_1,k_2)}(x). &(52)\end{aligned}$$

We will refer to (51), (52) as the outgoing and incoming singular value decompositions for the square $Y^{(l,k_1,k_2)}$, respectively.

We will be using finite-dimensional approximations to the operators (44), (45) obtained by truncating expressions (51), (52) after a finite number of terms. Specifically, given two natural numbers p_1 and r_1, we will define the operators

$$\begin{aligned} P_{p_1}^{(l,k_1,k_2)} &: L^2(Y^{(l,k_1,k_2)}) \to L^2(X^{(l,k_1,k_2)}), &(53)\\ R_{r_1}^{(l,k_1,k_2)} &: L^2(X^{(l,k_1,k_2)}) \to L^2(Y^{(l,k_1,k_2)}) &(54)\end{aligned}$$

by the formulae

$$(P_{p_1}^{(l,k_1,k_2)} \cdot \sigma)(x) = \int_{Y^{(l,k_1,k_2)}} K_{p_1}(x,y)\sigma(y)\,dy, \tag{55}$$

$$(R_{r_1}^{(l,k_1,k_2)} \cdot \sigma)(y) = \int_{X^{(l,k_1,k_2)}} K_{r_1}(y,x)\sigma(x)\,dx, \tag{56}$$

with

$$K_{p_1}(x,y) = \sum_{k=1}^{p_1} u_k^{out,(l,k_1,k_2)}(x) s_k^{out,(l,k_1,k_2)} v_k^{out,(l,k_1,k_2)}(y), \tag{57}$$

$$K_{r_1}(y,x) = \sum_{k=1}^{r_1} u_k^{in,(l,k_1,k_2)}(y) s_k^{in,(l,k_1,k_2)} v_k^{in,(l,k_1,k_2)}(x). \tag{58}$$

Substituting (57), (58) into (55), (56), we obtain

$$(P_{p_1}^{(l,k_1,k_2)} \cdot f)(x) = \sum_{k=1}^{p_1} u_k^{out,(l,k_1,k_2)}(x) s_k^{out,(l,k_1,k_2)} a_k^{out,(l,k_1,k_2)}, \tag{59}$$

with the coefficients $a_k^{out,(l,k_1,k_2)}$ given by the formula

$$a_k^{out,(l,k_1,k_2)} = \int_{Y^{(l,k_1,k_2)}} v_k^{out,(l,k_1,k_2)}(y)\sigma(y)\,dy, \tag{60}$$

and

$$(R_{r_1}^{(l,k_1,k_2)} \cdot \sigma)(y) = \sum_{k=1}^{r_1} u_k^{in,(l,k_1,k_2)}(y) s_k^{in,(l,k_1,k_2)} a_k^{in,(l,k_1,k_2)}, \tag{61}$$

with the coefficients $a_k^{in,(l,k_1,k_2)}$ given by the formula

$$a_k^{in,(l,k_1,k_2)} = \int_{X^{(l,k_1,k_2)}} v_k^{in,(l,k_1,k_2)}(x)\sigma(x)\,dx. \tag{62}$$

The function $(P_{p_1}^{(l,k_1,k_2)} \cdot \sigma) \in L^2(X^{(l,k_1,k_2)})$ with $\sigma \in L^2(Y^{(l,k_1,k_2)})$ will be referred to as the outgoing singular function expansion due to the charge distribution σ on the square $Y^{(l,k_1,k_2)}$. Similarly, the function $(R_{r_1}^{(l,k_1,k_2)} \cdot \sigma) \in L^2(X^{(l,k_1,k_2)})$ with $\sigma \in L^2(Y^{(l,k_1,k_2)})$ will be referred to as the incoming singular function expansion due to some charge distribution σ on $X^{(l,k_1,k_2)}$.

3.2 Singular Function Expansions of the Potentials

The following theorem provides a tool for approximating potentials produced by arbitrary charge distributions.

Theorem 3.1. *Suppose that the outgoing potential $g^{(l,k_1,k_2)} \in L^2(X^{(l,k_1,k_2)})$ is induced by the charge distribution $\sigma^{(l,k_1,k_2)} : L^2(Y^{(l,k_1,k_2)}) \to R$, i.e.*

$$g^{(l,k_1,k_2)}(x) = (P^{(l,k_1,k_2)} \cdot \sigma^{(l,k_1,k_2)})(x) = \int_{Y^{(l,k_1,k_2)}} K(x,y)\sigma^{(l,k_1,k_2)}(y)\,dy. \tag{63}$$

Then

$$g^{(l,k_1,k_2)}(x) = \sum_{k=1}^{\infty} u_k^{out,(l,k_1,k_2)}(x) s_k^{out,(l,k_1,k_2)} a_k^{out,(l,k_1,k_2)}, \tag{64}$$

with the coefficients $a_k^{out,(l,k_1,k_2)}$ *given by the formula*

$$a_k^{out,(l,k_1,k_2)} = \int_{Y^{(l,k_1,k_2)}} \sigma^{(l,k_1,k_2)}(y) v_k^{out,(l,k_1,k_2)}(y)\, dy. \tag{65}$$

Furthermore, for any $p \geq 1$,

$$\begin{aligned} &||g^{(l,k_1,k_2)}(x) - \sum_{k=1}^{p} u_k^{out,(l,k_1,k_2)}(x) s_k^{out,(l,k_1,k_2)} a_k^{out,(l,k_1,k_2)}||_{L^2(X^{(l,k_1,k_2)})} \leq \\ &\leq s_{p+1}^{out,(l,k_1,k_2)} ||\sigma^{(l,k_1,k_2)}||_{L^2(Y^{(l,k_1,k_2)})}, \end{aligned} \tag{66}$$

and

$$\sum_{k=1}^{p} |a_k^{out,(l,k_1,k_2)}|^2 \leq ||\sigma^{(l,k_1,k_2)}||^2_{L^2(Y^{(l,k_1,k_2)})}. \tag{67}$$

Proof. (66) follows immediately from Theorem 2.10. Singular values $s_k^{out,(l,k_1,k_2)}$ converge to zero as $k \to \infty$; therefore, (66) implies (64). Finally, due to (65), $a_k^{out,(l,k_1,k_2)}$ are the coefficients in the orthonormal basis $\{v_k^{out,(l,k_1,k_2)}\}$, from which (67) follows immediately. ☐

3.3 Translation Operators and Error Bounds

The following three theorems constitute the principal analytical tool for manipulating outgoing and incoming singular function expansions. Theorems 3.2, 3.4 provide formulae for the translation of outgoing and incoming singular function expansions, respectively. Theorem 3.3 describes a mechanism for converting an outgoing singular function expansion into an incoming singular function expansion.

Theorem 3.2 (Outgoing to Outgoing). *Suppose that the outgoing singular function expansion* $g^{out,(l,k_1,k_2)} : L^2(X^{(l,k_1,k_2)}) \to R$ *is given by the formula*

$$g^{out,(l,k_1,k_2)}(x) = \sum_{k=1}^{\infty} u_k^{out,(l,k_1,k_2)}(x) s_k^{out,(l,k_1,k_2)} a_k^{out,(l,k_1,k_2)}, \tag{68}$$

with the coefficients $a_k^{out,(l,k_1,k_2)}$ *such that*

$$\sum_{k=1}^{\infty} |a_k^{out,(l,k_1,k_2)}|^2 < +\infty, \tag{69}$$

and that $Y^{(l,k_1,k_2)} \subset Y^{(l-1,m_1,m_2)}$.

Then there exists a linear mapping

$$A^{(l-1,m_1,m_2),(l,k_1,k_2)} : l^2(N) \to l^2(N) \tag{70}$$

converting the sequence of coefficients $\{a_k^{out,(l,k_1,k_2)}\}$, $k = 1, 2, \ldots$ *into the sequence* $\{a_m^{out,(l-1,m_1,m_2)}\}$, $m = 1, 2, \ldots$, *defined by the formulae*

$$a_m^{out,(l-1,m_1,m_2)} = \sum_{k=1}^{\infty} A_{mk}^{(l-1,m_1,m_2),(l,k_1,k_2)} a_k^{out,(l,k_1,k_2)}, \tag{71}$$

$$A_{mk}^{(l-1,m_1,m_2),(l,k_1,k_2)} = \int_{Y^{(l,k_1,k_2)}} v_k^{out,(l,k_1,k_2)}(y) v_m^{out,(l-1,m_1,m_2)}(y)\, dy, \tag{72}$$

such that for all x *inside* $X^{(l-1,m_1,m_2)}$,

$$g^{out,(l,k_1,k_2)}(x) = \sum_{m=1}^{\infty} u_m^{out,(l-1,m_1,m_2)}(x) s_m^{out,(l-1,m_1,m_2)} a_m^{out,(l-1,m_1,m_2)}. \tag{73}$$

Furthermore, for any $p \geq 1$,

$$\begin{aligned} &\|g^{out,(l,k_1,k_2)}(x) - \sum_{m=1}^{p} u_m^{out,(l-1,m_1,m_2)}(x) s_m^{out,(l-1,m_1,m_2)} a_m^{out,(l-1,m_1,m_2)}\|_{L^2(X^{(l-1,m_1,m_2)})} \leq \\ &\leq s_{p+1}^{out,(l-1,m_1,m_2)} \sqrt{\sum_{k=1}^{\infty} |a_k^{out,(l,k_1,k_2)}|^2}. \end{aligned} \tag{74}$$

Proof. We observe that $g^{out,(l,k_1,k_2)}$ can be viewed as the potential

$$g^{out,(l,k_1,k_2)}(x) = \int_{Y^{(l,k_1,k_2)}} K(x,y) \sigma^{(l,k_1,k_2)}(y)\, dy \tag{75}$$

induced by the charge distribution $\sigma^{(l,k_1,k_2)} : L^2(Y^{(l,k_1,k_2)}) \to R$, defined by the formula

$$\sigma^{(l,k_1,k_2)}(y) = \sum_{k=1}^{\infty} a_k^{out,(l,k_1,k_2)} v_k^{out,(l,k_1,k_2)}(y). \tag{76}$$

We will denote by $\sigma^{(l-1,m_1,m_2)}$ the charge distribution on the square $Y^{(l-1,m_1,m_2)}$ given by the formula

$$\sigma^{(l-1,m_1,m_2)}(y) = \begin{cases} \sigma^{(l,k_1,k_2)}(y), & \text{if } y \in Y^{(l,k_1,k_2)}, \\ 0, & \text{if } y \in Y^{(l-1,m_1,m_2)} \backslash Y^{(l,k_1,k_2)}, \end{cases} \tag{77}$$

and by $g^{(l-1,m_1,m_2)}$ the outgoing potential on $X^{(l-1,m_1,m_2)}$ due to the distribution $\sigma^{(l-1,m_1,m_2)}$ on the square $Y^{(l-1,m_1,m_2)}$, i.e.,

$$\begin{aligned} g^{(l-1,m_1,m_2)}(x) &= (P^{(l-1,m_1,m_2)} \cdot \sigma^{(l-1,m_1,m_2)})(x) = \\ &= \int_{Y^{(l-1,m_1,m_2)}} K(x,y) \sigma^{(l-1,m_1,m_2)}(y)\, dy. \end{aligned} \tag{78}$$

Due to Theorem 3.1,

$$g^{(l-1,m_1,m_2)}(x) = \sum_{m=1}^{\infty} u_m^{out,(l-1,m_1,m_2)}(x) s_m^{out,(l-1,m_1,m_2)} a_m^{out,(l-1,m_1,m_2)}, \tag{79}$$

with the coefficients $a_m^{out,(l-1,m_1,m_2)}$ defined by the formula

$$a_m^{out,(l-1,m_1,m_2)} = \int_{Y^{(l-1,m_1,m_2)}} \sigma^{(l-1,m_1,m_2)}(y) v_m^{out,(l-1,m_1,m_2)}(y)\, dy. \tag{80}$$

Now, using (77), we have

$$a_m^{out,(l-1,m_1,m_2)} = \int_{Y^{(l,k_1,k_2)}} \sigma^{(l,k_1,k_2)}(y) v_m^{out,(l-1,m_1,m_2)}(y)\, dy. \tag{81}$$

Substituting (76) into (81), we arrive at

$$\begin{aligned} a_m^{out,(l-1,m_1,m_2)} &= \sum_{k=1}^{\infty} a_k^{out,(l,k_1,k_2)} \left(\int_{Y^{(l,k_1,k_2)}} v_k^{out,(l,k_1,k_2)}(y) v_m^{out,(l-1,m_1,m_2)}(y)\, dy \right) = \\ &= \sum_{k=1}^{\infty} a_k^{out,(l,k_1,k_2)} A_{mk}^{(l,k_1,k_2),(l-1,m_1,m_2)}, \end{aligned} \tag{82}$$

where

$$A_{mk}^{(l,k_1,k_2),(l-1,m_1,m_2)} = \int_{Y^{(l,k_1,k_2)}} v_k^{out,(l,k_1,k_2)}(y) v_m^{out,(l-1,m_1,m_2)}(y)\, dy. \tag{83}$$

Now, from the combination of (78) and Theorem 2.10, we obtain

$$\begin{aligned} &|| \int_{Y^{(l-1,m_1,m_2)}} K(x,y) \sigma^{(l-1,m_1,m_2)}(y)\, dy - \\ &\sum_{m=1}^{p} u_m^{out,(l-1,m_1,m_2)}(x) s_m^{out,(l-1,m_1,m_2)} a_m^{out,(l-1,m_1,m_2)} ||_{L^2(X^{(l-1,m_1,m_2)})} \leq \\ &\leq s_{p+1}^{out,(l-1,m_1,m_2)} ||\sigma^{(l-1,m_1,m_2)}||_{L^2(Y^{(l-1,m_1,m_2)})}. \end{aligned} \tag{84}$$

Due to (77), we have

$$\begin{aligned} &|| \int_{Y^{(l,k_1,k_2)}} K(x,y) \sigma^{(l,k_1,k_2)}(y)\, dy - \\ &\sum_{m=1}^{p} u_m^{out,(l-1,m_1,m_2)}(x) s_m^{out,(l-1,m_1,m_2)} a_m^{out,(l-1,m_1,m_2)} ||_{L^2(X^{(l-1,m_1,m_2)})} \leq \\ &\leq s_{p+1}^{out,(l-1,m_1,m_2)} ||\sigma^{(l,k_1,k_2)}||_{L^2(Y^{(l,k_1,k_2)})}. \end{aligned} \tag{85}$$

Thus,

$$\begin{aligned} &||g^{out,(l,k_1,k_2)}(x) - \sum_{m=1}^{p} u_m^{out,(l-1,m_1,m_2)}(x) s_m^{out,(l-1,m_1,m_2)} a_m^{out,(l-1,m_1,m_2)} ||_{L^2(X^{(l-1,m_1,m_2)})} \leq \\ &\leq s_{p+1}^{out,(l-1,m_1,m_2)} \sqrt{\sum_{k=1}^{\infty} |a_k^{out,(l,k_1,k_2)}|^2}. \end{aligned} \tag{86}$$

Finally, the singular values $s_k^{out,(l,k_1,k_2)}$ converge to zero as $k \to \infty$; therefore, (86) implies (73), and from the combination of (76), (77), (80), we have

$$\begin{aligned} &\sum_{m=1}^{p} |a_m^{out,(l-1,m_1,m_2)}|^2 \leq ||\sigma^{(l-1,m_1,m_2)}||^2_{L^2(Y^{(l-1,m_1,m_2)})} = \\ &= ||\sigma^{(l,k_1,k_2)}||^2_{L^2(Y^{(l,k_1,k_2)})} = \sum_{k=1}^{\infty} |a_k^{out,(l,k_1,k_2)}|^2. \end{aligned} \tag{87}$$

□

The proof of the following two theorems is virtually identical to that of Theorem 3.2, and is omitted.

Theorem 3.3 (Outgoing to Incoming). *Suppose that the outgoing singular function expansion* $g^{out,(l,k_1,k_2)} : L^2(X^{(l,k_1,k_2)}) \to R$ *is given by the formula*

$$g^{out,(l,k_1,k_2)}(x) = \sum_{k=1}^{\infty} u_k^{out,(l,k_1,k_2)}(x) s_k^{out,(l,k_1,k_2)} a_k^{out,(l,k_1,k_2)}, \tag{88}$$

with the real coefficients $a_k^{out,(l,k_1,k_2)}$ *such that*

$$\sum_{k=1}^{\infty} |a_k^{out,(l,k_1,k_2)}|^2 < +\infty, \tag{89}$$

and that $Y^{(l,m_1,m_2)} \subset X^{(l,k_1,k_2)}$.

Then there exists a linear mapping

$$B^{(l,m_1,m_2),(l,k_1,k_2)} : l^2(N) \to l^2(N) \tag{90}$$

converting the sequence of coefficients $\{a_k^{out,(l,k_1,k_2)}\}$, $k = 1, 2, \dots$ *into the sequence* $\{a_m^{in,(l,m_1,m_2)}\}$, $m = 1, 2, \dots,$ *defined by the formulae*

$$a_m^{in,(l,m_1,m_2)} = \sum_{k=1}^{\infty} B_{mk}^{(l,m_1,m_2),(l,k_1,k_2)} a_k^{out,(l,k_1,k_2)}, \tag{91}$$

$$B_{mk}^{(l,m_1,m_2),(l,k_1,k_2)} = \int_{Y^{(l,k_1,k_2)}} v_k^{out,(l,k_1,k_2)}(y) v_m^{in,(l,m_1,m_2)}(y)\, dy, \tag{92}$$

such that for all x *inside* $Y^{(l,m_1,m_2)}$,

$$g^{out,(l,k_1,k_2)}(x) = \sum_{m=1}^{\infty} u_m^{in,(l,m_1,m_2)}(x) s_m^{in,(l,m_1,m_2)} a_m^{in,(l,m_1,m_2)} \tag{93}$$

and

$$\sum_{m=1}^{\infty} |a_m^{in,(l,m_1,m_2)}|^2 \le \sum_{k=1}^{\infty} |a_k^{out,(l,k_1,k_2)}|^2. \tag{94}$$

Furthermore, for any $p \ge 1$,

$$||g^{out,(l,k_1,k_2)}(x) - \sum_{m=1}^{p} u_m^{in,(l,m_1,m_2)}(x) s_m^{in,(l,m_1,m_2)} a_m^{in,(l,m_1,m_2)}||_{L^2(Y^{(l,m_1,m_2)})} \le$$

$$\le s_{p+1}^{in,(l,m_1,m_2)} \sqrt{\sum_{k=1}^{\infty} |a_k^{out,(l,k_1,k_2)}|^2}. \tag{95}$$

Theorem 3.4 (Incoming to Incoming). *Suppose that the incoming singular function expansion* $g^{in,(l,k_1,k_2)} : L^2(Y^{(l,k_1,k_2)}) \to R$ *is given by the formula*

$$g^{in,(l,k_1,k_2)}(x) = \sum_{k=1}^{\infty} u_k^{in,(l,k_1,k_2)}(x) s_k^{in,(l,k_1,k_2)} a_k^{in,(l,k_1,k_2)}, \tag{96}$$

with the coefficients $a_k^{in,(l,k_1,k_2)}$ *such that*

$$\sum_{k=1}^{\infty} |a_k^{in,(l,k_1,k_2)}|^2 < +\infty, \tag{97}$$

and that $Y^{(l+1,m_1,m_2)} \subset Y^{(l,k_1,k_2)}$.

Then there exists a linear mapping

$$C^{(l+1,m_1,m_2),(l,k_1,k_2)} : l^2(N) \to l^2(N) \tag{98}$$

converting the sequence of coefficients $\{a_k^{in,(l,k_1,k_2)}\}$, $k = 1, 2, \ldots$ *into the sequence* $\{a_m^{in,(l+1,m_1,m_2)}\}$, $m = 1, 2, \ldots$, *defined by the formulae*

$$a_m^{in,(l,m_1,m_2)} = \sum_{k=1}^{\infty} C_{mk}^{(l+1,m_1,m_2),(l,k_1,k_2)} a_k^{out,(l,k_1,k_2)}, \tag{99}$$

where

$$C_{mk}^{(l+1,m_1,m_2),(l,k_1,k_2)} = \int_{X^{(l,k_1,k_2)}} v_k^{in,(l,k_1,k_2)}(y) v_m^{in,(l+1,m_1,m_2)}(y)\, dy, \tag{100}$$

such that for all y *inside* $Y^{(l+1,m_1,m_2)}$,

$$g^{in,(l,k_1,k_2)}(x) = \sum_{m=1}^{\infty} u_m^{in,(l+1,m_1,m_2)}(x) s_m^{in,(l+1,m_1,m_2)} a_m^{in,(l+1,m_1,m_2)} \tag{101}$$

and

$$\sum_{m=1}^{\infty} |a_m^{in,(l+1,m_1,m_2)}|^2 \le \sum_{k=1}^{\infty} |a_k^{in,(l,k_1,k_2)}|^2. \tag{102}$$

Furthermore, for any $p \ge 1$,

$$||g^{in,(l,k_1,k_2)}(x) - \sum_{m=1}^{p} u_m^{in,(l+1,m_1,m_2)}(x) s_m^{in,(l+1,m_1,m_2)} a_m^{in,(l+1,m_1,m_2)}||_{L^2(Y^{(l+1,m_1,m_2)})} \le$$

$$\le s_{p+1}^{in,(l+1,m_1,m_2)} \sqrt{\sum_{k=1}^{\infty} |a_k^{in,(l,k_1,k_2)}|^2}. \tag{103}$$

3.4 Singular Value Decompositions of Translation Operators

The algorithm of the following section (like its counterpart for harmonic fields described, for example, in [3]) depends on the efficient application of the translation operators (70), (90), (98) to arbitrary vectors. Clearly, these operators convert functions on the square into functions on the square, and could be extremely expensive to deal with numerically. Fortunately, Theorems 2.7, 2.8 of Section 2 guarantee that (asymptotically speaking) the cost of applying each of the operators (70), (90), (98) to an arbitrary vector is of the order

$$c + d \cdot \log(\varepsilon)^4, \tag{104}$$

with the constants c, d independent of the operator to be applied (as long as the conditions of Theorem 2.8 are satisfied). We will discuss the procedure for the efficient numerical evaluation of the operator (90) in some detail; the operators (70), (98) are in this respect identical to the operator (90).

Let us consider the operator (90) with some m_1, m_2, k_1, k_2. Choosing some natural n, we construct an $n \times n$ tensor-product Gaussian discretization of each of the squares $Y^{(l,m_1,m_2)}$, $Y^{(l,k_1,k_2)}$, and expand the kernel K on $Y^{(l,m_1,m_2)} \times Y^{(l,k_1,k_2)}$ into a 4-dimensional tensor product Legendre series. Due to Theorem 2.8, the error of such an expansion is bounded by

$$b(1+n)^4 \cdot q^n, \tag{105}$$

where b is a positive constant and $|q| < 1$. Choosing $n = c + d \cdot \log(\varepsilon)$, we guarantee that the error of our expansion is less than any arbitrary a-priori prescribed ε. An examination of (105) shows that the length of the expansion required to obtain reasonable accuracy is not excessive, though it is considerably greater than the lengths expansions required for harmonic kernels (see, for example, [3]). An additional improvement in the required lengths of expansions is obtained by replacing the tensor-product Legendre expansions of the operators (70), (90), (98) with their Singular Value Decompositions via Theorems 2.9, 2.10, 2.11. The cost of this latter step (in terms of CPU time requirements) is of the order p^3, and would be excessive, except for the fact that this procedure has to be performed only once for each kernel, since the necessary SVDs can be precomputed and stored; needless to say, this requires an amount of storage proportional to $p \cdot n^2$.

Remark 3.5. *The situation is simplified when the kernel K is convolutional, i.e depends only on the difference between its arguments. Indeed, in this case, the SDVs of the translation operators* $A^{(l-1,m_1,m_2),(l,k_1,k_2)}$, $B^{(l,m_1,m_2),(l,k_1,k_2)}$, $C^{(l+1,m_1,m_2),(l,k_1,k_2)}$ *do not have to be calculated for all interacting pairs of squares on all levels, but only for all interactions of a single square on each level. In this case, the construction of the SVDs requires trivial amounts of both CPU time and disk space. When the kernel K is not only convolutional but possesses additional symmetry (rotational, up-down, etc.) the situation is further simplified.*

4 Generalized Fast Multipole Method in Two Dimensions

4.1 Notation

In this section we will introduce the notation to be used in the description of the algorithm.

For any subset A of the computational box, $T(A)$ will denote the set of particles inside A.

B_l is the set of all nonempty boxes at the level l. B_0 will denote the computational box itself.

If box contains more than s particles, it is called a *parent* box. Otherwise, the box is said to be *childless.* Note that s is the maximum number of points in a childless box.

A *child* box is nonempty box obtained from the division of a parent box into four.

Colleagues are adjacent boxes of the same size at the same level. A given box has at most eight colleagues.

Two boxes b and c are said to be *well separated* if they are separated a distance greater or equal to the length of the size of the smallest box.

With each box b at the level l, we will associate five lists of other boxes.

List 1 of a box b will be denoted by U_b. It is empty if b is a parent box. If b is childless, it consists of b and of all childless boxes c that are adjacent to b.

List 2 of a box b will be denoted by V_b. It consists of all boxes c that are children of the colleagues of the b's parent and that are well separated from b.

List 3 of a box b will be denoted by W_b. It is empty if b is a parent box. If b is childless, it consists of all descendants of b's colleagues whose parent are adjacent to b but who are not adjacent to b themselves. Note that b is separated from each box c in W_b by a distance greater or equal to the length of the size of c.

List 4 of a box b will be denoted by X_b. It consists of all boxes c such that $b \in W_c$. Note that all boxes in List 4 are childless and larger that b.

List 5 of a box b will be denoted by Y_b. It consists of all boxes c that are well separated from b's parent.

Φ_b will denote the p-term outgoing singular function expansion for the box b.

Ψ_b will denote the p-term incoming singular function expansion for the box b.

Γ_b will denote the p-term incoming singular function expansion for the box b due to all particles in $T(V_b)$.

Δ_b will denote the p-term incoming singular function expansion for the box b due to all charges in $T(X_b)$.

$\Psi_b(r)$ is the result of evaluation of the expansion Ψ_b at a particle $r \in T(b)$.

$\alpha_b(r)$ will denote the potential at $r \in T(b)$ due to all particles in $T(U_b)$.

$\beta_b(r)$ will denote the potential at $r \in T(b)$ due to all particles in $T(W_b)$.

$\gamma_b(r)$ will denote the potential at $r \in T(b)$ due to all particles in $T(Y_b)$.

$F(r)$ will denote the potential at r.

$A_{b,c}$ will denote the translation operator (a $p \times p$ matrix) in the Theorem 3.2 for the boxes b and c such that $b = Y^{(l-1,m_1,m_2)}$ and $c = Y^{(l,k_1,k_2)}$.

$B_{b,c}$ will denote the translation operator (a $p \times p$ matrix) in the Theorem 3.3 for the boxes b and c such that $b = Y^{(l,m_1,m_2)}$ and $c = Y^{(l,k_1,k_2)}$.

$C_{b,c}$ will denote the translation operator (a $p \times p$ matrix) in the Theorem 3.4 for the boxes b and c such that $b = Y^{(l+1,m_1,m_2)}$ and $c = Y^{(l,k_1,k_2)}$.

4.2 Informal Description of the Algorithm

1. Create the adaptive quad-tree. Compute the outgoing and incoming singular functions for each box in the computational tree, by the means of the Theorem 2.11.

2. For each childless box b, the interactions between particles in $T(b)$ and $T(U_b)$ are evaluated directly. For each particle $r \in T(b)$ the result is $\alpha_b(r)$.

3. For each childless box b, form an outgoing singular function expansion Φ_b by the means of Theorem 3.1. For each parent box b, use Theorem 3.2 to translate and merge the outgoing singular function expansions of its children into the outgoing singular function expansion Φ_b.
4. Use Theorem 3.3 to convert the outgoing singular expansion of each box in V_b into the incoming singular function expansion in the box b, adding the resulting expansions together to obtain Γ_b.
5. Convert the potential of all particles in $T(X_b)$ into a incoming singular function expansion in the box b, adding the resulting expansions to obtain Δ_b. Add Δ_b to Γ_b.
6. For each childless box b, evaluate the potential $\beta_b(r)$ due to all particles in $T(W_b)$ by evaluating the outgoing singular function expansions Φ_c for each box $c \in W_b$.
7. Translate the incoming singular function expansion Γ_B of b's parent B to the box b by the means of Theorem 3.4. Add the resulting local expansion to Γ_b.
8. For each childless box b, evaluate the local expansion Γ_b at every particle $r \in b$ and add the result to $\alpha_b(r)$ and $\beta_b(r)$, obtaining the potential $F(r)$ at r.

4.3 Detailed Description of the Algorithm

Step 1: Initialization

Comment [Set the order n of Legendre expansions, the number of terms p in all singular function expansions, and the maximum number s of the particles in a childless box. Create the computational tree.]

```
do $l = 0, 1, 2, \dots$
    do $b \in B_l$
        if $b$ contains more than $s$ particles then
            subdivide $b$ into four smaller boxes,
            ignore empty boxes, add nonempty boxes to $B_{l+1}$.
        endif
    enddo
enddo
```

Comment [For each box b in the computational tree, compute the outgoing and incoming singular value decompositions of the kernel K.]

```
do $l = 0, 1, 2, \dots$
    do $b \in B_l$
        Set $b = Y^{(l,k_1,k_2)}$. Compute two singular value decompositions for $x \in$
        $X^{(l,k_1,k_2)}$, $y \in Y^{(l,k_1,k_2)}$.
```

```
            $$K(x,y) = \sum_{k=1}^{\infty} u_{b;k}^{out}(x) \cdot s_{b;k}^{out} \cdot v_{b;k}^{out}(y),$$
            $$K(y,x) = \sum_{k=1}^{\infty} u_{b;k}^{in}(y) \cdot s_{b;k}^{in} \cdot v_{b;k}^{in}(x).$$
    enddo
enddo
```

Step 2: Local Interactions

Comment [For each childless box b, evaluate interactions with the particles in $T(U_b)$ directly, obtaining the potential due to nearby particles.]

```
do $l = 0, 1, 2, \ldots$
    do $b \in B_l$, $b$ is childless
        do $x_i \in T(b), x_j \in T(U_b)$
            $$\alpha_b(x_i) = \alpha_b(x_i) + \sum_j q_j \cdot K(x_i, x_j).$$
        enddo
    enddo
enddo
```

Cost [$9(N/s) \cdot s \cdot s + 8(N/s) \cdot s \cdot s$ operations.]

Step 3: Outgoing Singular Function Expansions

Comment [For each childless box b, form the outgoing singular function expansion Φ_b.]

```
do $l = 0, 1, 2, \ldots$
    do $b \in B_l$, $b$ is childless
        Evaluate the coefficients of the outgoing singular function expansion for the
        square $b$ by the means of the Theorem 3.1.,
            $$\Phi_{b;k} = \sum_{x_j \in b} q_j \cdot v_{b;k}^{out}(x_j),$$
        for all $k = 1, \ldots, p$.
    enddo
enddo
```

Cost [Np operations.]

Step 4: Upward Sweep

Comment [For each parent box b, form the outgoing singular function expansion Φ_b by translating the outgoing singular function expansions of b's children and adding the resulting expansions together.]

```
do $l = \ldots, 2, 1, 0$
    do $b \in B_l$, $b$ is a parent box
        Use Theorem 3.2 to translate and merge the outgoing singular function expansions of $b's$ children $b_1$, $b_2$, $b_3$, $b_4$ into the outgoing singular function expansion $\Phi_b$

            $\Phi_b = \Phi_b + A_{b,b_1} \cdot \Phi_{b_1} + A_{b,b_2} \cdot \Phi_{b_2} + A_{b,b_3} \cdot \Phi_{b_3} + A_{b,b_4} \cdot \Phi_{b_4}$

    enddo
enddo
```

Cost [$(4/3)(N/s) \cdot p^2$ operations.]

Step 5: Adaptive Part

Comment [For each childless box b, form the incoming singular function expansion Δ_b due to particles located in List 4 of b.]

```
do $l = 0, 1, 2, \ldots$
    do $b \in B_l$, $b$ is childless
        Use Theorem 3.1 to evaluate the coefficients of the incoming singular function expansion $\Delta_b$ for the square $b$

            $\Delta_{b;k} = \sum_{x_i \in X_b} q_i \cdot v^{in}_{b;k}(x_i),$

        for all $k = 1 \ldots, p$.
    enddo
enddo
```

Cost [$8(N/s) \cdot p \cdot s$ operations.]

Comment [For each box b, evaluate the outgoing singular function expansion Φ_b at each particle located in boxes c in List 4 of b.]

```
do $l = 0, 1, 2, \ldots$
    do $b \in B_l$, $b$ is childless
        do $x_i \in X_b$
```

```
                $\beta_b(x_i) = \beta_b(x_i) + \sum_{k=1}^{p} \Phi_{b;k} \cdot s_{b;k}^{out} \cdot u_{b;k}^{out}(x_i).$
        enddo
    enddo
enddo
```

Cost [$8(N/s) \cdot p \cdot s$ operations.]

Step 6: Outgoing to Incoming

Comment [For each box b, convert the outgoing singular function expansion Φ_c for each box c in List 2 of b, into the incoming singular function expansion Γ_b, adding the resulting expansions together.]

```
do $l = 0, 1, 2, \ldots$
    do $b \in B_l$
        For all boxes $c \in V_b$, convert the outgoing singular function expansion into
        the incoming singular function expansion for the box $b$ by the means of
        Theorem 3.3. Add the resulting singular function expansions to $\Gamma_b$

                $\Gamma_b = \Gamma_b + \sum_{c \in V_b} B_{b,c} \cdot \Phi_c.$

    Add $\Gamma_b$ and $\Delta_b$ to obtain the incoming singular function expansion $\Psi_b$

                $\Psi_b = \Gamma_b + \Delta_b.$
    enddo
enddo
```

Cost [$27 \cdot (4/3)(N/s) \cdot p^2$ operations.]

Step 7: Downward Sweep

Comment [For every parent box b, translate the incoming singular function expansion Ψ_b to b's children incoming singular function expansions.]

```
do $l = 0, 1, 2, \ldots$
    do $b \in B_l$, $b$ is a parent box
        do $c \in B_{l+1}$, $c$ is a $b's$ child
            Translate the incoming singular function expansion $\Psi_b$ by the means of The-
            orem 3.4. Add the resulting local expansion to $\Psi_c$

                $\Psi_c = \Psi_c + C_{c,b} \cdot \Psi_b.$
        enddo
    enddo
enddo
```

Cost [$(4/3)(N/s) \cdot p^2$ operations.]

Step 8

Comment [For every childless box b, evaluate incoming singular function expansions Ψ_b at each particle, obtaining the potential due to distant particles. Find the potential at $r \in b$ by adding $\alpha_b(r)$, $\beta_b(r)$, $\gamma_b(r)$ together.]

```
do $l = 0, 1, 2, \ldots$
    do $b \in B_l$, $b$ is childless
        For each particle $x_j \in b$, evaluate
            $$\gamma_b(x_j) = \sum_{k=1}^{p} \Psi_{b;k} \cdot s_{b;k}^{in} \cdot u_{b;k}^{in}(x_j).$$
        Add $\alpha_b(x_j)$, $\beta_b(x_j)$, $\gamma_b(x_j)$ to obtain the potential $F(x_j)$ at $x_j \in b$
            $$F(x_j) = \alpha_b(x_j) + \beta_b(x_j) + \gamma_b(x_j).$$
    enddo
enddo
```

Cost [$N \cdot p$ operations.]

4.4 Complexity of the Algorithm

Since s is the average number of particles in a childless box at the finest level, there are approximately N/s childless boxes, and approximately

$$B = (1 + 1/4 + 1/4^2 + \ldots) \cdot (N/s) = \frac{4}{3} \cdot \frac{N}{s} \tag{106}$$

boxes in the tree hierarchy. Therefore, Step 3 requires Np work, Step 4 requires Bp^2 work, Step 6 requires $27Bp^2$ work, Step 7 requires Bp^2 work, Step 8 requires Np work, and Step 2 requires $9 \cdot N/s \cdot s \cdot s = 9Ns$ work. Thus, a reasonable estimate for the total operation count is

$$9Ns + 2Np + 29Bp^2 = 9Ns + 2Np + 29 \cdot \frac{4}{3} \cdot \frac{N}{s} \cdot p^2. \tag{107}$$

With $s = 2p$, the operation count becomes approximately

$$40\,Np. \tag{108}$$

```
                $\beta_b(x_i) = \beta_b(x_i) + \sum_{k=1}^{p} \Phi_{b:k} \cdot s_{b:k}^{out} \cdot u_{b:k}^{out}(x_i).$
        enddo
    enddo
enddo
```

Cost [$8(N/s) \cdot p \cdot s$ operations.]

Step 6: Outgoing to Incoming

Comment [For each box b, convert the outgoing singular function expansion Φ_c for each box c in List 2 of b, into the incoming singular function expansion Γ_b, adding the resulting expansions together.]

```
do $l = 0, 1, 2, \ldots$
    do $b \in B_l$
        For all boxes $c \in V_b$, convert the outgoing singular function expansion into
        the incoming singular function expansion for the box $b$ by the means of
        Theorem 3.3. Add the resulting singular function expansions to $\Gamma_b$
            $\Gamma_b = \Gamma_b + \sum_{c \in V_b} B_{b,c} \cdot \Phi_c.$
    Add $\Gamma_b$ and $\Delta_b$ to obtain the incoming singular function expansion $\Psi_b$
            $\Psi_b = \Gamma_b + \Delta_b.$
    enddo
enddo
```

Cost [$27 \cdot (4/3)(N/s) \cdot p^2$ operations.]

Step 7: Downward Sweep

Comment [For every parent box b, translate the incoming singular function expansion Ψ_b to b's children incoming singular function expansions.]

```
do $l = 0, 1, 2, \ldots$
    do $b \in B_l$, $b$ is a parent box
        do $c \in B_{l+1}$, $c$ is a $b's$ child
            Translate the incoming singular function expansion $\Psi_b$ by the means of Theorem 3.4. Add the resulting local expansion to $\Psi_c$
                $\Psi_c = \Psi_c + C_{c,b} \cdot \Psi_b.$
```

digits, we used 90-term singular function expansions, and obtained these (during the precomputation stage) by starting with Legendre expansions of order 16.

2. For the 3-digit version of the scheme, the break-even point with the direct scheme is $n \sim 200$; for 6 digits, the break-even point is $n \sim 800$, and for 10-digits the scheme becomes faster than the direct one at $n \sim 3000$.

3. The efficiency of the algorithm does not suffer significantly when the charges in the simulation are clustered. On the other hand, unlike its counterpart for harmonic kernels, the algorithm of this paper does not seem to derive any advantage from the clustering of particles in the simulation.

4. The cost of the algorithm grows rapidly with the increase of accuracy requirements. The algorithm is considerably slower than modern versions of the FMM for harmonic fields, especially in high-accuracy environments (see, for example, [10]).

5.1 Generalizations and Conclusions

The algorithm of this paper has an obvious analogue in three dimensions: quad-trees are replaced with oct-trees, two-dimensional expansions are replaced with three-dimensional ones, and the programming becomes more involved. Such a scheme has been implemented (see [6]), and found to work satisfactorily, as long as the required precision is low. For accuracies better than three or four digits, the CPU time requirements of the three-dimensional scheme become excessive.

For many kernels, the algorithm of this paper can be accelerated via an approach similar to the one used by [4], [9], [10] to accelerate the FMM for harmonic fields in two and three dimensions. Specifically, most the operators (70), (90), (98) can be diagonalized; this requires that the kernel K be approximated by linear combinations of exponentials on appropriately chosen parts of the product $Y^{(l,k_1,k_2)} \times X^{(l,k_1,k_2)}$. Needless to say, this can not be done for a "general" kernel K; however, it appears to be possible for many kernels (and classes of kernels) of interest. Such a scheme would require several developments (both analytic and numerical); it would accelerate the two-dimensional version of the algorithm significantly. The real pay-off of such a project would be in three dimensions, where it would be likely to make large-scale high-precision simulations feasible.

6 Acknowledgments

We would like to thank Professor Leslie Greengard for many useful discussions and suggestions.

The adaptive part of the algorithm in the Step 5 requires $O(8(N/s)ps + 8(N/s)ps) = O(16Np)$ work, and Step 3 requires additional $O(8(N/s)s^2) = O(8Ns)$ work. The total operation count is

$$17Ns + 18Np + 29Bp^2 = 17Ns + 18Np + 29(4/3)(N/s)p^2. \tag{109}$$

By setting $s = 1.5p$, the operation count becomes approximately

$$69\,Np. \tag{110}$$

5 Numerical Results

A FORTRAN program has been written implementing the algorithm described in the preceding section. All timings listed below correspond to calculations performed on an UltraSparc-I/167 computer with 128MB RAM, using double precision arithmetic. The order of Legendre expansions was $n = 4$, $n = 8$, and $n = 16$ and the number of singular functions varied from $p = 9$ to $p = 36$ to $p = 90$ in order to achieve roughly 3, 6 and 10 digits accuracy, respectively.

The results of these experiments are presented in the tables below. The first column contains the number of particles used in the simulation. The second column contains the time for construction of the computational tree and precomputation of values singular functions at locations of particles. This can be done once for any given configuration of particles. We do not include the time for precomputation of singular value decompositions in this column, since this can be done in advance for any given kernel. The third column contains the total run time of the algorithm. The fourth and the fifth columns contain the actual time required by the algorithm and the time required by the direct algorithm, respectively.

Finally, the last two columns contain the relative 2-norm E_2 and the relative maximum error E_∞ obtained at any one particle. They are defined by the formulae

$$E_2 = \left(\frac{\sum_{i=1}^{N} |f_i - \tilde{f}_i|^2}{\sum_{i=1}^{N} |f_i|^2} \right)^{1/2}, \quad E_\infty = \max_i \frac{|f_i - \tilde{f}_i|}{|f_i|}, \tag{111}$$

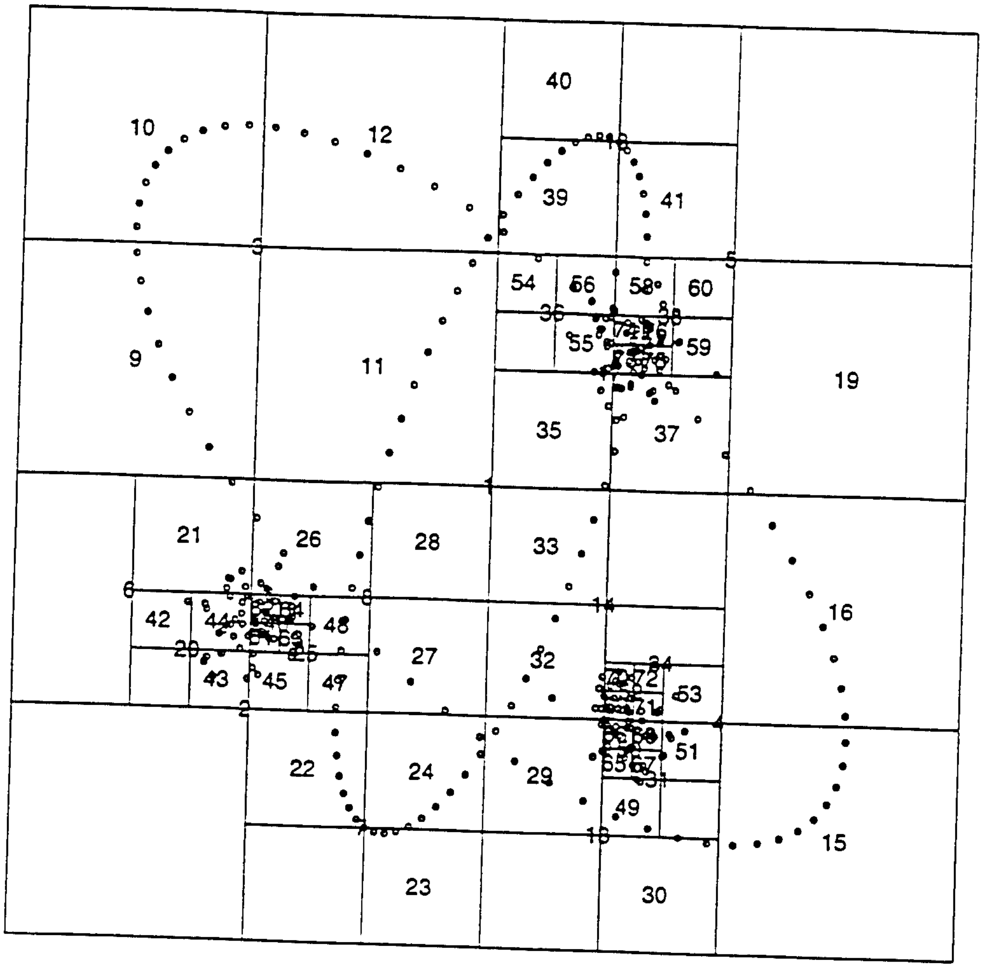
where f_i is the value of the potential at the i-th particle position obtained by the direct calculation, and $\tilde{f}_i$ is the result obtained by the algorithm.

For the first set of tests, the positions of particles were uniformly distributed in the unit square. For the second set of tests, two fifth of charged particles were distributed uniformly along two ellipses and the remaining of particles were distributed randomly in three circles with a gaussian density. The number of terms in the singular function expansions was set to 9, 36 and 90, and the number of particles in a childless box was set to 15, 61, and 153, respectively.

Several observations can be made from Tables 1–12 below, and from the more extensive numerical experiments performed by the authors.

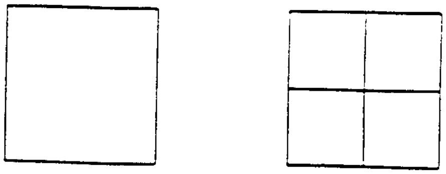
1. The number of singular functions required to obtain 3-digit accuracy is 9; the corresponding order of the Legendre expansions is 4. The 6-digit scheme requires 36-term singular-function expansions, and Legendre expansions of order 8. In order to obtain 10

Figure 2: Non-uniform distribution of charges and its associated adaptive quad-tree.



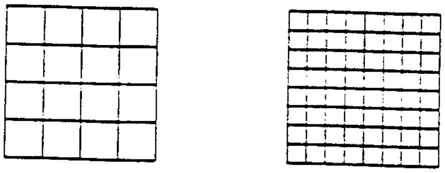
10
12
40
39
41
3
5
54
56
58
60
36
38
55
59
9
11
19
35
37
1
21
26
28
33
6
8
14
16
42
44
48
20
27
32
43
45
47
34
72
53
2
22
24
29
51
49
13
15
23
30

Figure 1: The computational box and three levels of refinement.



Level 0

Level 1



Level 2

Level 3

N	$T_{init}(s)$	$T_{alg}(s)$	$T_{run}(s)$	$T_{dir}(s)$	E_2	E_∞
200	0.007	0.009	0.015	0.019	0.11770E-03	0.85266E-03
400	0.015	0.018	0.034	0.076	0.27390E-03	0.19749E-02
800	0.024	0.047	0.071	0.310	0.29473E-03	0.20307E-02
1600	0.062	0.089	0.151	1.344	0.39506E-03	0.36146E-02
3200	0.105	0.213	0.318	5.371	0.42503E-03	0.38485E-02
6400	0.266	0.399	0.666	21.783	0.49194E-03	0.43736E-02

Table 1: Uniformly distributed particles. $K(x, y) = 1/|x - y|$, $s = 15$, $p = 9$, and $n = 4$.

N	$T_{init}(s)$	$T_{alg}(s)$	$T_{run}(s)$	$T_{dir}(s)$	E_2	E_∞
400	0.066	0.042	0.107	0.075	0.37968E-07	0.36455E-06
800	0.124	0.130	0.254	0.309	0.30664E-07	0.23301E-06
1600	0.255	0.251	0.505	1.347	0.59016E-07	0.63131E-06
3200	0.492	0.684	1.176	5.375	0.67426E-07	0.67145E-06
6400	0.997	1.230	2.227	21.756	0.16065E-06	0.16568E-05

Table 2: Uniformly distributed particles. $K(x, y) = 1/|x - y|$, $s = 61$, $p = 36$, and $n = 8$.

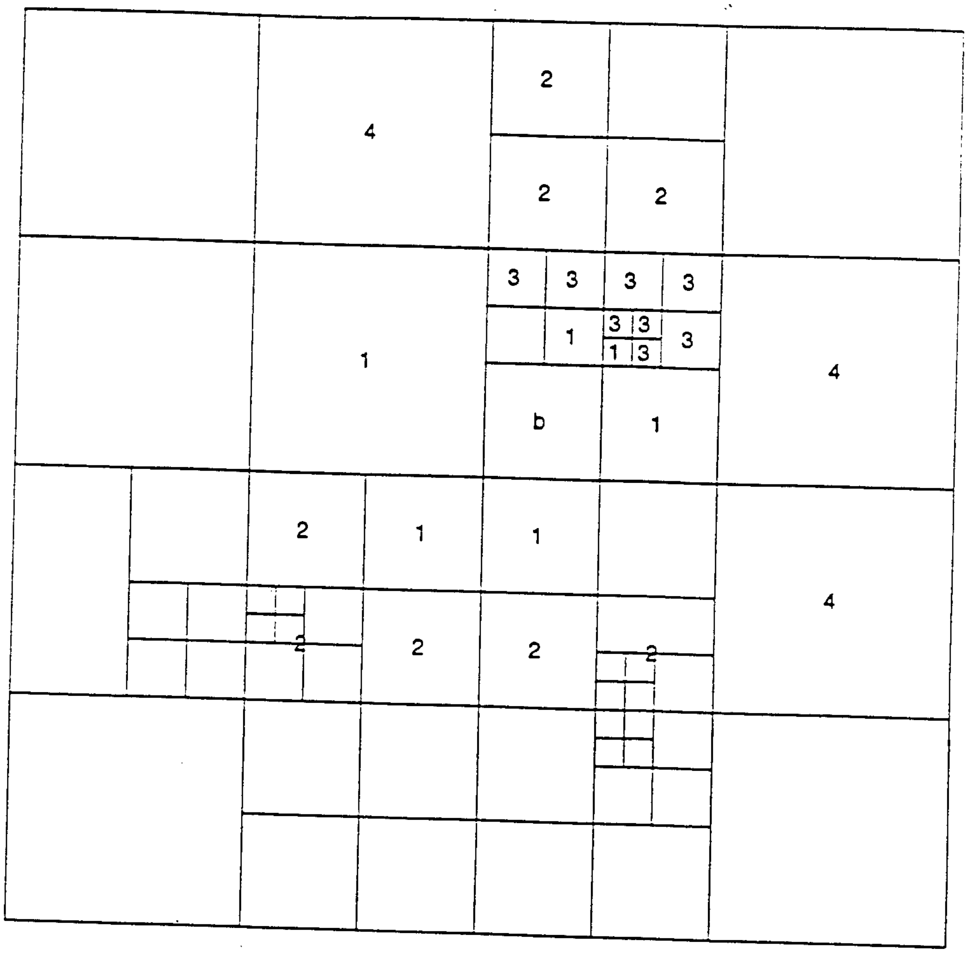
N	$T_{init}(s)$	$T_{alg}(s)$	$T_{run}(s)$	$T_{dir}(s)$	E_2	E_∞
800	0.832	0.213	1.045	0.316	0.35519E-11	0.27597E-10
1600	1.625	0.580	2.205	1.342	0.27911E-11	0.23206E-10
3200	3.210	1.374	4.515	5.371	0.47909E-11	0.35374E-10
6400	6.301	3.138	9.438	21.798	0.40687E-11	0.47116E-10

Table 3: Uniformly distributed particles. $K(x, y) = 1/|x - y|$, $s = 153$, $p = 90$, and $n = 16$.

N	$T_{init}(s)$	$T_{alg}(s)$	$T_{run}(s)$	$T_{dir}(s)$	E_2	E_∞
200	0.007	0.007	0.014	0.014	0.33680E-06	0.11237E-02
400	0.015	0.016	0.031	0.055	0.24487E-05	0.46567E-02
800	0.024	0.037	0.061	0.227	0.75789E-05	0.67792E-02
1600	0.063	0.077	0.140	1.016	0.36380E-04	0.82441E-02
3200	0.105	0.173	0.278	4.064	0.10114E-03	0.11347E-01
6400	0.267	0.353	0.619	16.397	0.42311E-04	0.12510E-01

Table 4: Uniformly distributed particles. $K(x, y) = 1/|x - y|^2$, $s = 15$, $p = 9$, and $n = 4$.

Figure 3: Box *b* and its associated Lists 1 to 4 for the charge distribution in Figure 2.



2
4
2
2
3
3
3
3
3
3
1
1
3
3
1
4
b
1
2
1
1
4
2
2
2
2

N	$T_{init}(s)$	$T_{alg}(s)$	$T_{run}(s)$	$T_{dir}(s)$	E_2	E_∞
800	0.826	0.180	1.006	0.231	0.14987E-12	0.17505E-09
1600	1.597	0.450	2.047	1.009	0.32363E-12	0.74589E-10
3200	3.205	1.217	4.422	4.104	0.20036E-11	0.25330E-09
6400	6.315	2.507	8.823	16.404	0.46900E-12	0.16662E-09

Table 6: Uniformly distributed particles. $K(x, y) = 1/|x-y|^2$, $s = 153$, $p = 90$, and $n = 16$.

Figure 5: Highly non-uniformly distributed particles and the associated partition of the computational box.

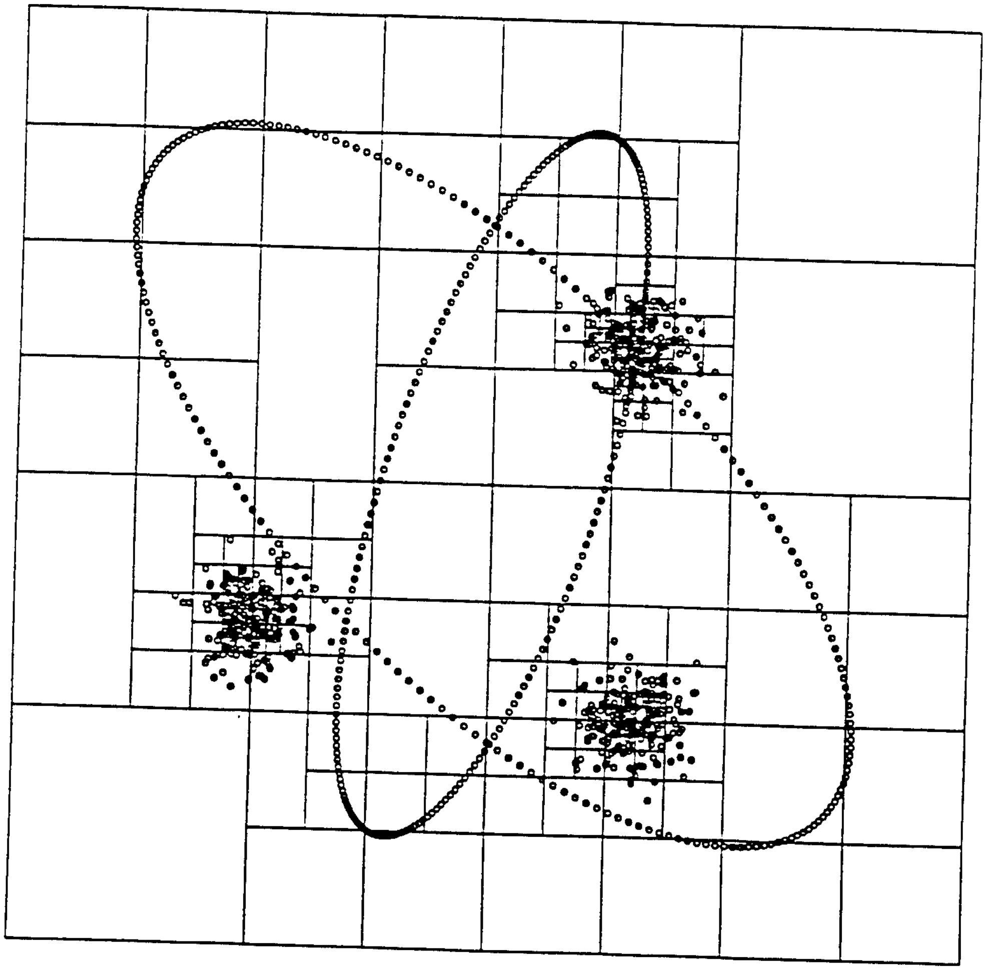
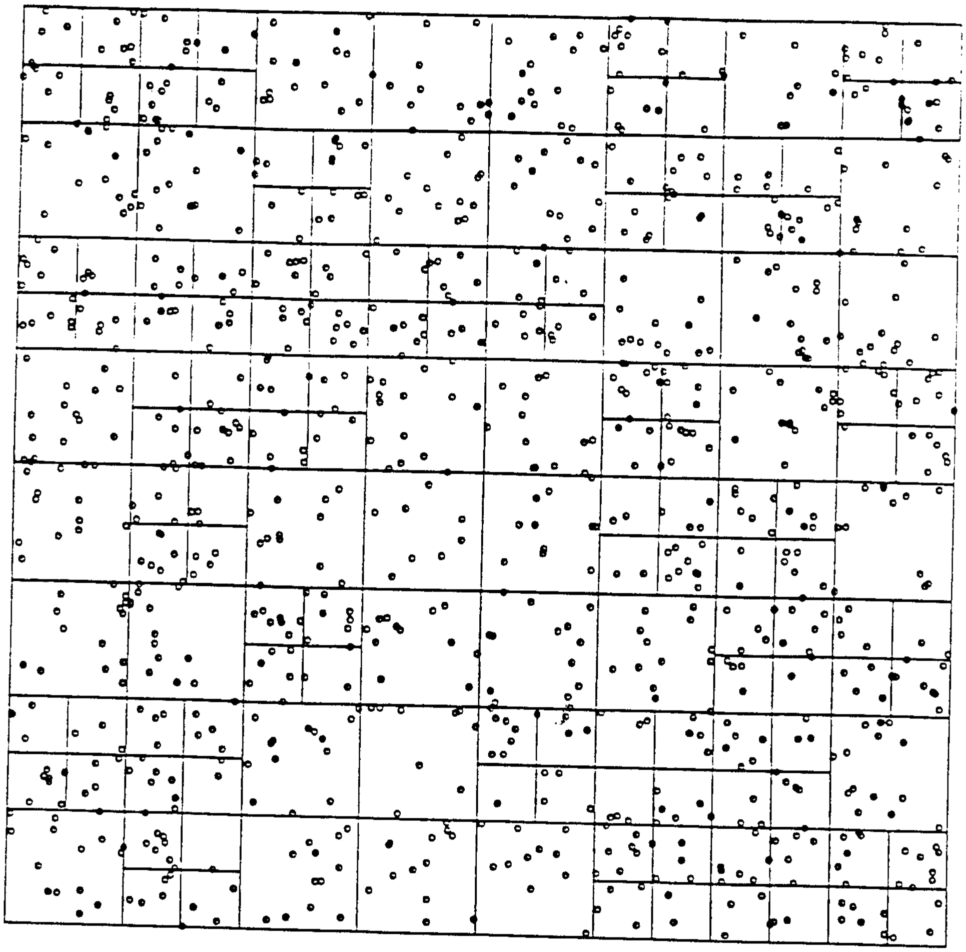


Figure 4: Uniformly distributed particles and the associated partition of the computational box.



N	$T_{init}(s)$	$T_{alg}(s)$	$T_{run}(s)$	$T_{dir}(s)$	E_2	E_∞
400	0.065	0.033	0.099	0.055	0.33610E-09	0.96336E-06
800	0.126	0.098	0.223	0.225	0.74619E-09	0.55977E-06
1600	0.254	0.210	0.465	1.016	0.59034E-08	0.21584E-05
3200	0.493	0.529	1.022	4.090	0.18124E-07	0.17612E-05
6400	0.996	1.036	2.031	16.365	0.14692E-07	0.47616E-05

Table 5: Uniformly distributed particles. $K(x,y) = 1/|x-y|^2$, $s = 61$, $p = 36$, and $n = 8$.

N	$T_{init}(s)$	$T_{alg}(s)$	$T_{run}(s)$	$T_{dir}(s)$	E_2	E_∞
400	0.016	0.023	0.039	0.055	0.44531E-04	0.19765E-02
800	0.029	0.045	0.075	0.226	0.72969E-04	0.37896E-02
1600	0.058	0.100	0.158	1.016	0.98016E-04	0.70910E-02
3200	0.115	0.197	0.312	4.064	0.24054E-03	0.57700E-02
6400	0.225	0.382	0.608	16.405	0.23213E-03	0.82506E-02

Table 10: Highly non-uniformly distributed particles. $K(x, y) = 1/|x - y|^2$, $s = 15$, $p = 9$, and $n = 4$.

N	$T_{init}(s)$	$T_{alg}(s)$	$T_{run}(s)$	$T_{dir}(s)$	E_2	E_∞
400	0.065	0.045	0.110	0.054	0.61825E-08	0.15019E-05
800	0.140	0.108	0.247	0.234	0.10608E-07	0.20936E-05
1600	0.265	0.312	0.577	1.016	0.13661E-07	0.18906E-05
3200	0.521	0.639	1.160	4.059	0.38933E-07	0.21694E-05
6400	1.043	1.439	2.481	16.408	0.38956E-07	0.61407E-05

Table 11: Highly non-uniformly distributed particles. $K(x, y) = 1/|x - y|^2$, $s = 61$, $p = 36$, and $n = 8$.

N	$T_{init}(s)$	$T_{alg}(s)$	$T_{run}(s)$	$T_{dir}(s)$	E_2	E_∞
800	0.805	0.192	0.996	0.230	0.10539E-11	0.41111E-09
1600	1.717	0.477	2.194	1.010	0.68055E-12	0.18332E-09
3200	3.338	1.352	4.691	4.144	0.28719E-11	0.39139E-09
6400	6.540	4.045	10.586	16.411	0.29936E-11	0.21587E-09

Table 12: Highly non-uniformly distributed particles. $K(x, y) = 1/|x - y|^2$, $s = 153$, $p = 90$, and $n = 16$.

N	$T_{init}(s)$	$T_{alg}(s)$	$T_{run}(s)$	$T_{dir}(s)$	E_2	E_∞
200	0.008	0.010	0.019	0.019	0.13524E-03	0.87697E-03
400	0.016	0.029	0.045	0.076	0.20754E-03	0.11468E-02
800	0.029	0.058	0.087	0.309	0.26133E-03	0.12042E-02
1600	0.057	0.126	0.183	1.344	0.32551E-03	0.26410E-02
3200	0.114	0.245	0.358	5.368	0.37247E-03	0.34192E-02
6400	0.224	0.475	0.699	21.788	0.42360E-03	0.35911E-02

Table 7: Highly non-uniformly distributed particles. $K(x, y) = 1/|x - y|$, $s = 15$, $p = 9$, and $n = 4$.

N	$T_{init}(s)$	$T_{alg}(s)$	$T_{run}(s)$	$T_{dir}(s)$	E_2	E_∞
400	0.065	0.060	0.125	0.076	0.59124E-07	0.61426E-06
800	0.140	0.139	0.279	0.315	0.77114E-07	0.11068E-05
1600	0.264	0.413	0.677	1.336	0.10049E-06	0.97051E-06
3200	0.528	0.834	1.363	5.439	0.12151E-06	0.12184E-05
6400	1.052	1.867	2.919	21.761	0.15353E-06	0.15668E-05

Table 8: Highly non-uniformly distributed particles. $K(x, y) = 1/|x - y|$, $s = 61$, $p = 36$, and $n = 8$.

N	$T_{init}(s)$	$T_{alg}(s)$	$T_{run}(s)$	$T_{dir}(s)$	E_2	E_∞
800	0.805	0.250	1.055	0.314	0.40445E-11	0.87339E-10
1600	1.716	0.603	2.319	1.338	0.61795E-11	0.75092E-10
3200	3.334	1.769	5.103	5.442	0.88132E-11	0.85507E-10
6400	6.540	5.366	11.906	21.810	0.11716E-10	0.12124E-09

Table 9: Highly non-uniformly distributed particles. $K(x, y) = 1/|x - y|$, $s = 153$, $p = 90$, and $n = 16$.

[17] N. Yarvin and V. Rokhlin, *Generalized Gaussian Quadratures and Singular Value Decomposition of Integral Operators*, SIAM J. Sci. Stat. Comput., 20 (1998), 699–718.

[18] N. Yarvin and V. Rokhlin, *An Improved Fast Multipole Algorithm for Potential Fields on the Line*, SIAM J. Numer. Anal., 36 (1999), 629–666.

References

[1] M. ABRAMOWITZ AND I. STEGUN, *Handbook of Mathematical Functions*, Applied Mathematics Series, National Bureau of Standards, Washington, DC, 1964.

[2] B. K. ALPERT AND V. ROKHLIN, *A Fast Algorithm for the Evaluation of Legendre Expansions*, SIAM J. Sci. Stat. Comput., 12 (1991), 158–179.

[3] J. CARRIER, L. GREENGARD, AND V. ROKHLIN, *A Fast Adaptive Multipole Algorithm for Particle Simulations*, SIAM J. Sci. Stat. Comput., 9 (1988), 669–686.

[4] H. CHENG, L. GREENGARD AND V. ROKHLIN, *A Fast Multipole Algorithm in Three Dimensions*, J. Comput. Phys., 155 (1999), 468–498.

[5] A. DUTT, M. GU, AND V. ROKHLIN, *Fast Algorithms for Polynomial Interpolation, Integration and Differentiation*, SIAM J. Numer. Anal., 33 (1996), 1689–1711.

[6] Z. GIMBUTAS, *Ph.D. Dissertation*, Yale University, 1999.

[7] L. GREENGARD AND V. ROKHLIN, *A Fast Algorithm for Particle Simulations*, J. Comput. Phys., 73 (1987), 325–348.

[8] L. GREENGARD, *The Rapid Evaluation of Potential Fields in Particle Systems*, MIT Press, Cambridge, Mass., 1988.

[9] L. GREENGARD AND V. ROKHLIN, *A New version of the Fast Multipole Method for the Laplace Equation in Three Dimensions*, Acta Numerica 6 (1997), 229–269.

[10] T. HRYCAK AND V. ROKHLIN, *An Improved Fast Multipole Algorithm for Potential Fields*, SIAM J. Sci. Stat. Comput., 19 (1998), 1804–1826.

[11] S. KAPUR AND D. E. LONG, *IES3: Efficient Electrostatic and Electromagnetic Simulation*, IEEE Comput. Sci. Eng. 5 (1998), 60–67.

[12] M. J. D. POWELL, *Approximation Theory and Methods*, Cambridge University Press, 1981.

[13] J. R. PHILLIPS AND J. K. WHITE, *A Precorrected-FFT Method for Electrostatic Analysis of Complicated 3-D Structures*, IEEE Trans. Comput. Aid. D. 16 (1997), 1059–1072.

[14] V. ROKHLIN, *Rapid Solution of Integral Equations of Classical Potential Theory*, J. Comput. Phys., 60 (1986), 187–207.

[15] M. REED AND B. SIMON, *Methods of Modern Mathematical Physics*, Vol. 1, Academic Press, 1980.

[16] G. SZEGÖ, *Orthogonal Polynomials*, American Mathematical Society, Providence, 1978.

$$K_\gamma^{0,1}(\sigma)(s) = \int_0^L \frac{\partial \Phi_{\gamma(t)}(\gamma(s))}{\partial N(s)} \sigma(t)\, dt \,. \tag{137}$$

$$K_\gamma^{1,1}(\sigma)(s) = \text{f.p.} \int_0^L \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s))}{\partial N(s)\, \partial N(t)} \sigma(t)\, dt \,, \tag{138}$$

$$K_\gamma^{2,1}(\sigma)(s) = \text{f.p.} \int_0^L \frac{\partial^3 \Phi_{\gamma(t)}(\gamma(s))}{\partial N(s)\, \partial N(t)^2} \sigma(t)\, dt \,. \tag{139}$$

$$K_\gamma^{0,2}(\sigma)(s) = \text{f.p.} \int_0^L \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s))}{\partial N(s)^2} \sigma(t)\, dt \,, \tag{140}$$

$$K_\gamma^{1,2}(\sigma)(s) = \text{f.p.} \int_0^L \frac{\partial^3 \Phi_{\gamma(t)}(\gamma(s))}{\partial N(s)^2\, \partial N(t)} \sigma(t)\, dt \,, \tag{141}$$

$$K_\gamma^{0,3}(\sigma)(s) = \text{f.p.} \int_0^L \frac{\partial^3 \Phi_{\gamma(t)}(\gamma(s))}{\partial N(s)^3} \sigma(t)\, dt \,, \tag{142}$$

respectively.

Remark 3.4 *Obviously, the operators $K_\gamma^{0,1}$, $K_\gamma^{0,2}$, $K_\gamma^{0,3}$, $K^{1,2}$ given by the formulae (137), (140) - (142) are the adjoints of the operators $K_\gamma^{1,0}$, $K_\gamma^{2,0}$, $K_\gamma^{3,0}$, $K_\gamma^{2,1}$ defined by (134) - (136), (139). Furthermore, $K_\gamma^{1,1}$, defined by (138), is self-adjoint.*

4 Proof of Results

In this section we prove the results from Section 2. The outline of this section is as follows: First, we consider the case where γ is a circle. We provide the proof for Theorem 2.6. In Lemma 4.2 we give explicit formulas for the boundary integral operators (134) - (140) for the case where γ is a circle. Then, by combining Theorem 2.6 and Lemma 4.2, we immediately get the so-called jump conditions for the operators (12) - (25) on a circle. These are stated in Theorem 4.3.

Next, we consider the case where γ is an arbitrary and sufficiently smooth Jordan curve. Since the proof of the identities (94) - (99) in Theorem 2.8 are similar, we only provide the proof for (94) and (95). In fact, (94) and (95) in Theorem 2.8 follow immediately from Theorem 4.7 and Lemma 4.6. The proof of Theorem 4.7 is based on Theorem 4.3 and the approximation (178) given in Lemma 4.5.

Proof of Theorem 2.6 Since the proofs for the identities (50) - (64) are nearly identical, we only provide the proof for the interior limit of the quadruple layer potential (53). Further, it is enough to prove (53) for the case $r = 1$; the general case follows by a simple transformation of variables. We choose the parametrization

$$\gamma(t) = (\cos(t), \sin(t))\,, \tag{143}$$

where $t \in [-\pi, \pi]$. It immediately follows from (143) that

$$\int_{-\pi}^{\pi} \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s) - h \cdot N(s))}{\partial N(t)^2} e^{ikt}\, dt =$$

$$= \int_{-\pi}^{\pi} \frac{1 - 2 \cdot (1-h) \cdot \cos(t-s) + (1-h)^2 \cdot \cos(2(t-s))}{(1 + (1-h)^2 - 2 \cdot (1-h) \cdot \cos(t-s))^2} e^{ikt}\, dt$$

$$= e^{iks} \cdot \int_{-\pi}^{\pi} \frac{1 - 2 \cdot (1-h) \cdot \cos(t) + (1-h)^2 \cdot \cos(2t)}{(1 + (1-h)^2 - 2 \cdot (1-h) \cdot \cos(t))^2} e^{ikt}\, dt\,, \tag{144}$$

for any $s \in [-\pi, \pi]$. We will use calculus of residues to evaluate the integral (144). To this effect, the substitution

$$z = e^{it}\,, \tag{145}$$

converts (144) into

$$e^{iks} \cdot \int_{-\pi}^{\pi} \frac{1 - 2 \cdot (1-h) \cdot \cos(t) + (1-h)^2 \cdot \cos(2t)}{(1 + (1-h)^2 - 2 \cdot (1-h) \cdot \cos(t))^2} e^{ikt}\, dt =$$

$$= e^{iks} \cdot \int_{|z|=1} \frac{-i}{z} \left(\frac{1 - (1-h)(z + z^{-1}) + \frac{1}{2}(1-h)^2 (z^2 + z^{-2})}{(1 + (1-h)^2 - (1-h)(z + z^{-1}))^2} \right) \cdot z^k\, dz\,, \tag{146}$$

and after simple algebraic manipulation, we get

$$\frac{-i}{z} \left(\frac{1 - (1-h)(z + z^{-1}) + \frac{1}{2}(1-h)^2 (z^2 + z^{-2})}{(1 + (1-h)^2 - (1-h)(z + z^{-1}))^2} \right) \cdot z^k =$$

$$= \frac{1}{2} \cdot \left(-\frac{iz^{k+1}}{((1-h) - z)^2} - \frac{iz^{k-1}}{(z(1-h) - 1)^2} \right)\,. \tag{147}$$

Substituting (147) into (146), we obtain

$$\int_{-\pi}^{\pi} \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s) - h \cdot N(s))}{\partial N(t)^2} e^{ikt}\, dt =$$

$$= e^{iks} \cdot \int_{|z|=1} \frac{1}{2} \cdot \left(-\frac{iz^{k+1}}{((1-h) - z)^2} - \frac{iz^{k-1}}{(z(1-h) - 1)^2} \right) dz\,. \tag{148}$$

Now, formula (53) for $r = 1$ follows by applying a standard residue calculation to (148). □

Remark 4.1 *Formulae (50) - (52), (57) - (58) follow from well-known results (see for example [11, 3]). While the derivation of (53) - (56), (59) - (64) is quite similar, the authors failed to find them in the literature.*

The operators $K_\gamma^{1,0}$, $K_\gamma^{2,0}$, $K_\gamma^{3,0}$, $K_\gamma^{1,1}$, $K_\gamma^{2,1}$, $K_\gamma^{0,1}$, $K_\gamma^{0,2}$, $K_\gamma^{0,3}$, $K_\gamma^{1,2}$ defined by (134) - (141), assume a particularly simple form on the circle. The following lemma follows immediately from an elementary computation.

Lemma 4.2 *Suppose that γ is a circle of radius r parametrized by its arclength with exterior unit normal denoted by N. Then, for any sufficiently smooth function $\sigma : [-\pi r, \pi r] \to \mathbb{C}$:*

$$(a) \qquad K_\gamma^{1,0}(\sigma)(s) = \int_{-\pi r}^{\pi r} -\frac{\sigma(t)}{2\,r}\,dt = -\pi\,\widehat{\sigma}_0\,, \tag{149}$$

$$\begin{aligned}(b) \qquad K_\gamma^{2,0}(\sigma)(s) &= \text{f.p.}\int_{-\pi r}^{\pi r}\left(\frac{1}{2\,r^2} + \frac{1}{2\,r^2\,\cos(\frac{t-s}{r}) - 2\,r^2}\right)\sigma(t)\,dt \\ &= \pi\,r^{-1}\,\widehat{\sigma}_0 + \pi\,H(\sigma')(s)\,,\end{aligned} \tag{150}$$

$$\begin{aligned}(c) \qquad K_\gamma^{3,0}(\sigma)(s) &= \text{f.p.}\int_{-\pi r}^{\pi r}\left(-\frac{1}{r^3} - \frac{3}{2\,r^3\,\cos(\frac{t-s}{r}) - 2\,r^3}\right)\sigma(t)\,dt \\ &= -2\,\pi\,r^{-2}\,\widehat{\sigma}_0 - 3\,\pi\,r^{-1}\,H(\sigma')(s)\,,\end{aligned} \tag{151}$$

$$(d) \qquad K_\gamma^{0,1}(\sigma)(s) = \int_{-\pi r}^{\pi r} -\frac{\sigma(t)}{2\,r}\,dt = -\pi\,\widehat{\sigma}_0\,, \tag{152}$$

$$(e) \qquad K_\gamma^{1,1}(\sigma)(s) = \text{f.p.}\int_{-\pi r}^{\pi r}\frac{\sigma(t)}{2\,r^2 - 2\,r^2\,\cos(\frac{t-s}{r})}\,dt = -\pi\,H(\sigma')(s)\,, \tag{153}$$

$$(f) \qquad K_\gamma^{2,1}(\sigma)(s) = \text{f.p.}\int_{-\pi r}^{\pi r}\frac{\sigma(t)}{2\,r^3\,\cos(\frac{t-s}{r}) - 2\,r^3}\,dt = \pi\,r^{-1}\,H(\sigma')(s)\,, \tag{154}$$

$$\begin{aligned}(g) \qquad K_\gamma^{0,2}(\sigma)(s) &= \text{f.p.}\int_{-\pi r}^{\pi r}\left(\frac{1}{2\,r^2} + \frac{1}{2\,r^2\,\cos(\frac{t-s}{r}) - 2\,r^2}\right)\sigma(t)\,dt \\ &= \pi\,r^{-1}\,\widehat{\sigma}_0 + \pi\,H(\sigma')(s)\,,\end{aligned} \tag{155}$$

$$(h) \qquad K_\gamma^{1,2}(\sigma)(s) = \text{f.p.}\int_{-\pi r}^{\pi r}\frac{\sigma(t)}{2\,r^3\,\cos(\frac{t-s}{r}) - 2\,r^3}\,dt = \pi\,r^{-1}\,H(\sigma')(s)\,, \tag{156}$$

$$\begin{aligned}(i) \qquad K_\gamma^{0,3}(\sigma)(s) &= \text{f.p.}\int_{-\pi r}^{\pi r}\left(-\frac{1}{r^3} - \frac{3}{2\,r^3\,\cos(\frac{t-s}{r}) - 2\,r^3}\right)\sigma(t)\,dt \\ &= -2\,\pi\,r^{-2}\,\widehat{\sigma}_0 - 3\,\pi\,r^{-1}\,H(\sigma')(s)\,,\end{aligned} \tag{157}$$

where H denotes the Hilbert transform (see (130) in Section 3.3).

The following theorem is an immediate consequence of Theorem 2.6 and Lemma 4.2. It summarizes the so-called jump conditions for the integrals (12) - (29) on the boundary Γ, where Γ is a circle.

Theorem 4.3 *Suppose that γ is a circle of radius r parametrized by its arclength with exterior unit normal denoted by N. Further, suppose that H denotes the Hilbert transform (130). Then, for any sufficiently smooth function $\sigma : [-\pi r, \pi r] \to \mathbb{C}$,*

$$(a) \qquad K_{\gamma,\mathrm{i}}^{1,0}(\sigma)(s) = -\pi\,\sigma(s) + K_\gamma^{1,0}(\sigma)(s)\,, \tag{158}$$

$$K^{1,0}_{\gamma,\mathrm{e}}(\sigma)(s) = \pi\,\sigma(s) + K^{1,0}_{\gamma}(\sigma)(s)\,, \tag{159}$$

$$\textit{(b)}\qquad K^{2,0}_{\gamma,\mathrm{i}}(\sigma)(s) = \pi\,r^{-1}\,\sigma(s) + K^{2,0}_{\gamma}(\sigma)(s)\,, \tag{160}$$

$$K^{2,0}_{\gamma,\mathrm{e}}(\sigma)(s) = -\pi\,r^{-1}\,\sigma(s) + K^{2,0}_{\gamma}(\sigma)(s)\,, \tag{161}$$

$$\textit{(c)}\qquad K^{3,0}_{\gamma,\mathrm{i}}(\sigma)(s) = -2\,\pi\,r^{-2}\,\sigma(s) + \pi\,\sigma''(s) + K^{3,0}_{\gamma}(\sigma)(s)\,. \tag{162}$$

$$K^{3,0}_{\gamma,\mathrm{e}}(\sigma)(s) = 2\,\pi\,r^{-2}\,\sigma(s) - \pi\,\sigma''(s) + K^{3,0}_{\gamma}(\sigma)(s)\,. \tag{163}$$

$$\textit{(d)}\qquad K^{0,1}_{\gamma,\mathrm{i}}(\sigma)(s) = \pi\,\sigma(s) + (K^{1,0}_{\gamma})^{\star}(\sigma)(s)\,, \tag{164}$$

$$K^{0,1}_{\gamma,\mathrm{e}}(\sigma)(s) = -\pi\,\sigma(s) + (K^{1,0}_{\gamma})^{\star}(\sigma)(s)\,, \tag{165}$$

$$\textit{(e)}\qquad K^{1,1}_{\gamma,\mathrm{i}}(\sigma)(s) = K^{1,1}_{\gamma,\mathrm{e}}(\sigma)(s) = K^{1,1}_{\gamma}(\sigma)(s) = -\pi\,H(\sigma')(s)\,, \tag{166}$$

$$\textit{(f)}\qquad K^{2,1}_{\gamma,\mathrm{i}}(\sigma)(s) = -\pi\,\sigma''(s) + K^{2,1}_{\gamma}(\sigma)(s)\,, \tag{167}$$

$$K^{2,1}_{\gamma,\mathrm{e}}(\sigma)(s) = \pi\,\sigma''(s) + K^{2,1}_{\gamma}(\sigma)(s)\,, \tag{168}$$

$$\textit{(g)}\qquad K^{0,2}_{\gamma,\mathrm{i}}(\sigma)(s) = -\pi\,r^{-1}\,\sigma(s) + K^{0,2}_{\gamma}(\sigma)(s)\,, \tag{169}$$

$$K^{0,2}_{\gamma,\mathrm{e}}(\sigma)(s) = \pi\,r^{-1}\,\sigma(s) + K^{0,2}_{\gamma}(\sigma)(s)\,, \tag{170}$$

$$\textit{(h)}\qquad K^{1,2}_{\gamma,\mathrm{i}}(\sigma)(s) = \pi\,\sigma''(s) + (K^{2,1}_{\gamma})^{\star}(\sigma)(s)\,, \tag{171}$$

$$K^{1,2}_{\gamma,\mathrm{e}}(\sigma)(s) = -\pi\,\sigma''(s) + (K^{2,1}_{\gamma})^{\star}(\sigma)(s)\,, \tag{172}$$

$$\textit{(i)}\qquad K^{0,3}_{\gamma,\mathrm{i}}(\sigma)(s) = 2\,\pi\,r^{-2}\,\sigma(s) - \pi\,\sigma''(s) + (K^{3,0}_{\gamma})^{\star}(\sigma)(s)\,, \tag{173}$$

$$K^{0,3}_{\gamma,\mathrm{e}}(\sigma)(s) = -2\,\pi\,r^{-2}\,\sigma(s) + \pi\,\sigma''(s) + (K^{3,0}_{\gamma})^{\star}(\sigma)(s)\,. \tag{174}$$

We now proceed to the case where γ is an arbitrary sufficiently smooth Jordan curve. The following obvious lemma can be found in most elementary textbooks on differential geometry (see, for example, [4]).

Lemma 4.4 *Suppose that $\gamma : [0, L] \to \mathbb{R}^2$ is a sufficiently smooth Jordan curve parametrized by its arclength with the exterior unit normal and the unit tangent vectors at $\gamma(s)$ denoted by $N(s)$ and $T(s)$, respectively. Then, there exist a positive real number a (dependent on γ), and two continuously differentiable functions $f, g : (-a, a) \to \mathbb{R}$ (dependent on γ), such that for any $s \in [0, L]$,*

$$\gamma(s+t) - \gamma(s) = \Big(t + t^3 \cdot f(t)\Big) \cdot T(s) - \left(\frac{c\,t^2}{2} + t^3 \cdot g(t)\right) \cdot N(s)\,, \tag{175}$$

for all $t \in (-a, a)$, where the coefficient c in (175) is the curvature of γ at the point $\gamma(s)$. Furthermore, for all $t \in (-a, a)$,

$$|f(t)| \le \|\,\gamma'''(s)\|\,, \tag{176}$$

$$|g(t)| \le \|\,\gamma'''(s)\|\,. \tag{177}$$

In the local parametrization (175), the potential of a quadrupole located at $\gamma(s)$ and oriented in the direction $N(s)$ assumes a particularly simple form, given by the following lemma.

Lemma 4.5 *Suppose that $\gamma : [0, L] \to \mathbb{R}^2$ is a sufficiently smooth Jordan curve parametrized by its arclength. Then, there exist real positive numbers A, a and h_0 such that for any $s \in [0, L]$*

$$\left| \frac{\partial^2 \Phi_{\gamma(s+t)}(\gamma(s) - h \cdot N(s))}{\partial N(s+t)^2} - \frac{h^2 - t^2}{(h^2 + t^2)^2} - \frac{c\,h\,t^2\,(5\,h^2 - t^2)}{(h^2 + t^2)^3} \right| \le A\,. \tag{178}$$

for all $t \in (-a, a)$, $0 \le h < h_0$, where the coefficient c in (178) is the curvature of γ at the point $\gamma(s)$.

Proof. Without loss of generality, it is sufficient to prove the lemma for the case where $s = 0$, $\gamma(0) = 0$, and $\gamma'(0) = (1, 0)$. Substituting (175) into (9) and evaluating the result at $x = (0, h)$, we obtain

$$\frac{\partial^2 \Phi_{\gamma(t)}(x)}{\partial N(t)^2} = \frac{p_0(h, t)}{(h^2 + t^2 + r(h, t))^2}\,, \tag{179}$$

where $p_0, r : \mathbb{R}^2 \to \mathbb{R}$ are functions given by the formulae

$$\begin{aligned}
p_0(h, t) \;=\; & \Big[h - t + c\,h\,t + \frac{c\,t^2}{2} - \frac{c^2\,t^3}{2} + 3\,h\,t^2\big(f(t) + g(t)\big) - 2\,t^3\big(2\,f(t) - g(t)\big) \\
& - \frac{c\,t^4}{2}\big(f(t) + 5\,g(t)\big) + h\,t^3\big(f'(t) + g'(t)\big) - t^4\big(f'(t) - g'(t)\big) - 3\,t^5\big(f(t)^2 + g(t)^2\big) \\
& - \frac{c\,t^5}{2}\big(f'(t) + g'(t)\big) - t^6\,f(t)\big(f'(t) - g'(t)\big) - t^6\,g(t)\big(f'(t) + g'(t)\big)\Big] \\
& \cdot \Big[h + t - c\,h\,t + \frac{c\,t^2}{2} + \frac{c^2\,t^3}{2} + 3\,h\,t^2\big(f(t) - g(t)\big) + 2\,t^3\big(2\,f(t) + g(t)\big) \\
& - \frac{c\,t^4}{2}\big(f(t) - 5\,g(t)\big) + h\,t^3\big(f'(t) - g'(t)\big) + t^4\big(f'(t) + g'(t)\big) + 3\,t^5\big(f(t)^2 + g(t)^2\big) \\
& - \frac{c\,t^5}{2}\big(f'(t) - g'(t)\big) + t^6\,f(t)\big(f'(t) + g'(t)\big) - t^6\,g(t)\big(f'(t) - g'(t)\big)\Big]\,, \qquad (180) \\
r(h, t) \;=\; & -c\,h\,t^2 - 2\,h\,t^3\,g(t) + \frac{c^2\,t^4}{4} + 2\,t^4\,f(t) + c\,t^5\,g(t) + t^6\big(f(t)^2 + g(t)^2\big)\,. \qquad (181)
\end{aligned}$$

We also introduce the notation

$$p_1(h, t) = \big(h^2 + t^2 + r(h, t)\big)^2 - \big(h^2 + t^2\big)^2 = 2\,(h^2 + t^2) \cdot r(h, t) + r(h, t)^2\,. \tag{182}$$

Obviously, (180) - (182) are algebraic combinations of f, g, f', g', t, and h, and an examination of formulae (180) - (182) immediately shows that there exist positive real numbers a, h_0, and

C (dependent on γ) such that

$$
\begin{aligned}
\left| p_0(h,t) - h^2 + t^2 - 3\,c\,h\,t^2 \right| &\le C(h^2 + t^2)^2, \qquad (183)\\
\left| p_0(h,t)\cdot p_1(h,t) - 2\,c\,h\,t^2\,(h^2+t^2)\,(h^2-t^2) \right| &\le C\,(h^2 + t^2)^4, \qquad (184)\\
\left| p_0(h,t)\cdot p_1(h,t)^2 \right| &\le C\,(h^2 + t^2)^6, \qquad (185)\\
\left| \frac{p_1(h,t)}{(h^2+t^2)^2} \right| &< 1, \qquad (186)
\end{aligned}
$$

for all $h < h_0$, $t \in (-a,a)$. Substituting (182) into (179), we have

$$
\begin{aligned}
\frac{\partial^2 \Phi_{\gamma(t)}(x)}{\partial N(t)^2} &= \frac{p_0(h,t)}{(h^2+t^2)^2\left(1+\frac{p_1(h,t)}{(h^2+t^2)^2}\right)}\\
&= \frac{p_0(h,t)}{(h^2+t^2)^2}\sum_{k=0}^{\infty}(-1)^k\,\frac{p_1(h,t)^k}{(h^2+t^2)^{2k}}, \qquad (187)
\end{aligned}
$$

where the convergence of the series follows from (186). Combining (183) – (185), we obtain

$$
\begin{aligned}
&\left| \frac{\partial^2 \Phi_{\gamma(t)}(x)}{\partial N(t)^2} - \frac{h^2-t^2}{(h^2+t^2)^2} - \frac{c\,h\,t^2\,(5\,h^2+t^2)}{(h^2+t^2)^3} \right| \le \left| \frac{p_0(h,t) - h^2 + t^2 - 3\,c\,h\,t^2}{(h^2+t^2)^2} \right|\\
&\quad + \left| \frac{p_0(h,t)\cdot p_1(h,t) - 2\,c\,h\,t^2\,(h^2+t^2)\,(h^2-t^2)}{(h^2+t^2)^4} \right| + \sum_{k=2}^{\infty}\left| \frac{p_0(h,t)\cdot p_1(h,t)^k}{(h^2+t^2)^{2k+2}} \right|\\
&\le 2\,C + C\cdot\frac{\alpha^2}{1-\alpha}, \qquad (188)
\end{aligned}
$$

with α defined by the formula

$$
\alpha = \sup_{h<h_0,\, t\in(-a,a)} \left| \frac{p_1(h,t)}{(h^2+t^2)^2} \right|. \qquad (189)
$$

Now, introducing the notation

$$
A = 2\,C + C\cdot\frac{\alpha^2}{1-\alpha}, \qquad (190)
$$

we obtain (178). □

Lemma 4.2 provides an explicit formula for the operator $K_\gamma^{2,0}$, defined in (135), in the case when γ is a circle. The following lemma shows that the operator $K_\gamma^{2,0}$ on an arbitrary sufficiently smooth Jordan curve of length L, is a compact perturbation of $K_\gamma^{2,0}$ on the circle of radius $\frac{L}{2\pi}$. Its proof is an immediate consequence of estimate (178) in Lemma 4.5.

Lemma 4.6 *Suppose that $\gamma : [0,L] \to \mathbb{R}^2$ is a sufficiently smooth Jordan curve parametrized by its arclength, and that $\eta : [0,L] \to \mathbb{R}^2$ denotes the circle of radius $\frac{L}{2\pi}$, also parametrized* *by its arclength. In addition, suppose that $\sigma : [0, L] \to \mathbb{R}$ is a twice continuously differentiable function. Then,*

$$\text{f.p.} \int_0^L \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s))}{\partial N(t)^2} \sigma(t)\, dt = \text{f.p.} \int_0^L \frac{\partial^2 \Phi_{\eta(t)}(\eta(s))}{\partial N(t)^2} \sigma(t)\, dt + M_2(\sigma)(s)\,, \tag{191}$$

where $M_2 : c[0, L] \to c[0, L]$ is a compact operator defined by the formula

$$M_2(\sigma)(s) = \int_0^L \left(\frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s))}{\partial N(t)^2} - \frac{\partial^2 \Phi_{\eta(t)}(\eta(s))}{\partial N(t)^2} \right) \sigma(t)\, dt\,. \tag{192}$$

Furthermore, for any $t \neq s$,

$$\begin{aligned} m_2(s, t) \quad &= \quad \frac{2\,\langle N(t), \gamma(s) - \gamma(t)\rangle^2}{\|\gamma(s) - \gamma(t)\|^4} - \frac{1}{2}\left(\frac{2\,\pi}{L}\right)^2 \\ &+ \frac{\|\gamma(s) - \gamma(t)\|^2 - 2\left(\frac{L}{2\pi}\right)^2 \left(1 - \cos\left(\frac{2\pi}{L}(s - t)\right)\right)}{\|\gamma(s) - \gamma(t)\|^2\, 2\left(\frac{L}{2\pi}\right)^2 \left(1 - \cos\left(\frac{2\pi}{L}(s - t)\right)\right)}\,, \end{aligned} \tag{193}$$

and for $t = s$,

$$m_2(s, s) = \frac{5}{12}\left(c(s)\right)^2 - \frac{5}{12}\left(\frac{2\,\pi}{L}\right)^2\,, \tag{194}$$

where $c(s)$ is the curvature of γ at the point $\gamma(s)$, and $m_2 : [0, L] \times [0, L] \to \mathbb{R}$ is the kernel of the operator M_2.

The following theorem provides the so-called jump conditions for the operators (14) and (15) on the boundary Γ, when Γ is sufficiently smooth.

Theorem 4.7 *Suppose that $\gamma : [0, L] \to \mathbb{R}^2$ is a sufficiently smooth Jordan curve parametrized by its arclength. Then, for any sufficiently smooth function $\sigma : [0, L] \to \mathbb{R}$,*

$$\begin{aligned} &K^{2,0}_{\gamma,\mathrm{e}}(\sigma)(s) - K^{2,0}_{\gamma,\mathrm{i}}(\sigma)(s) = \\ &\quad = \lim_{h \to 0} \int_0^L \left(\frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s) + h \cdot N(s))}{\partial N(t)^2} - \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s) - h \cdot N(s))}{\partial N(t)^2} \right) \sigma(t)\, dt \\ &\quad = -2\,\pi\, c(s)\, \sigma(s)\,, \end{aligned} \tag{195}$$

and

$$\begin{aligned} &K^{2,0}_{\gamma,\mathrm{e}}(\sigma)(s) + K^{2,0}_{\gamma,\mathrm{i}}(\sigma)(s) = \\ &\quad = \lim_{h \to 0} \int_0^L \left(\frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s) + h \cdot N(s))}{\partial N(t)^2} + \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s) - h \cdot N(s))}{\partial N(t)^2} \right) \sigma(t)\, dt \\ &\quad = 2 \cdot \text{f.p.} \int_0^L \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s))}{\partial N(t)^2} \sigma(t)\, dt\,, \end{aligned} \tag{196}$$

where $c(s)$ denotes the curvature of γ at $\gamma(s)$. In other words, the quadruple layer potential with density σ (see (6)), can be continuously extended from Ω to $\overline{\Omega}$ and from $\mathbb{R}^2 \setminus \overline{\Omega}$ to $\mathbb{R}^2 \setminus \Omega$, with the limiting values given by the formulae

$$p^{2,0}_{\gamma,\sigma,\mathrm{i}}(s) = K^{2,0}_{\gamma,\mathrm{i}}(\sigma)(s) = \pi\, c(s)\, \sigma(s) + \text{f.p.} \int_0^L \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s))}{\partial N(t)^2}\, \sigma(t)\, dt\,, \tag{197}$$

$$p^{2,0}_{\gamma,\sigma,\mathrm{e}}(s) = K^{2,0}_{\gamma,\mathrm{e}}(\sigma)(s) = -\pi\, c(s)\, \sigma(s) + \text{f.p.} \int_0^L \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s))}{\partial N(t)^2}\, \sigma(t)\, dt\,. \tag{198}$$

Proof. Without loss of generality, we can assume that $s \neq 0$ and $s \neq L$. We begin by proving (196). Suppose that $\eta : [0, L] \to \mathbb{R}^2$ is the circle of radius $\frac{L}{2\pi}$ parametrized by its arclength. We define the functions $\Sigma^h_\gamma, \Sigma^h_\eta : [0, L] \times [0, L] \to \mathbb{R}$ via the formulae

$$\Sigma^h_\gamma(s,t) = \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s) + h \cdot N(s))}{\partial N(t)^2} + \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s) - h \cdot N(s))}{\partial N(t)^2}\,, \tag{199}$$

$$\Sigma^h_\eta(s,t) = \frac{\partial^2 \Phi_{\eta(t)}(\eta(s) + h \cdot N(s))}{\partial N(t)^2} + \frac{\partial^2 \Phi_{\eta(t)}(\eta(s) - h \cdot N(s))}{\partial N(t)^2}\,, \tag{200}$$

and, substituting (199), (200) into (196), obtain the identity

$$K^{2,0}_{\gamma,\mathrm{e}}(\sigma)(s) + K^{2,0}_{\gamma,\mathrm{i}}(\sigma)(s) = \lim_{h\to 0} \int_0^L \Sigma^h_\eta(s,t)\, \sigma(t)\, dt + \lim_{h\to 0} \int_0^L \left(\Sigma^h_\gamma(s,t) - \Sigma^h_\eta(s,t)\right) \sigma(t)\, dt\,. \tag{201}$$

Substituting (160), (161) in Theorem 4.3 into (201), we have

$$\begin{aligned} &K^{2,0}_{\gamma,\mathrm{e}}(\sigma)(s) + K^{2,0}_{\gamma,\mathrm{i}}(\sigma)(s) = \\ &= 2 \cdot \text{f.p.} \int_0^L \frac{\partial^2 \Phi_{\eta(t)}(\eta(s))}{\partial N(t)^2}\, \sigma(t)\, dt + \lim_{h\to 0} \int_0^L \left(\Sigma^h_\gamma(s,t) - \Sigma^h_\eta(s,t)\right) \sigma(t)\, dt\,. \end{aligned} \tag{202}$$

Due to Lemma 4.5, there exist positive real constants C_0, a, and h_0 such that for any $s \in [0, L]$

$$\left| \Sigma^h_\gamma(s,t) - \Sigma^h_\eta(s,t) \right| \leq C_0\,, \tag{203}$$

for all $|t - s| < a$, $0 \leq h < h_0$. For any $t \neq s$ and sufficiently small h, both $\Sigma^h_\gamma(s,t)$ and $\Sigma^h_\eta(s,t)$ are c^∞-functions. Therefore, there also exist positive real constants h_1, C_1 such that for any $s \in [0, L]$

$$\left| \Sigma^h_\gamma(s,t) - \Sigma^h_\eta(s,t) \right| \leq C_1\,, \tag{204}$$

for all $|t - s| > a$, $0 \leq h < h_1$. Now, applying Lebesgue's dominated convergence theorem (see, for example, [18]) to the second integral of the right hand side of (202), we obtain

$$\begin{aligned} &\lim_{h\to 0} \int_0^L \left(\Sigma^h_\gamma(s,t) - \Sigma^h_\eta(s,t)\right) \sigma(t)\, dt = \\ &\quad= \int_0^L \lim_{h\to 0} \left(\Sigma^h_\gamma(s,t) - \Sigma^h_\eta(s,t)\right) \sigma(t)\, dt \\ &\quad= 2 \cdot \int_0^L \left(\frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s))}{\partial N(t)^2} - \frac{\partial^2 \Phi_{\eta(t)}(\eta(s))}{\partial N(t)^2} \right) \sigma(t)\, dt\,. \end{aligned} \tag{205}$$

Finally, formula (196) immediately follows from the combination of (202), (205) with (191), (192) in Lemma 4.6.

We now proceed by proving formula (195). We define the functions $\Delta^h_\gamma, \Delta^h_\eta : [0, L] \times [0, L] \to \mathbb{R}$ via the formulae

$$\Delta^h_\gamma(s,t) = \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s) + h \cdot N(s))}{\partial N(t)^2} - \frac{\partial^2 \Phi_{\gamma(t)}(\gamma(s) - h \cdot N(s))}{\partial N(t)^2}, \tag{206}$$

$$\Delta^h_\eta(s,t) = \frac{\partial^2 \Phi_{\eta(t)}(\eta(s) + h \cdot N(s))}{\partial N(t)^2} - \frac{\partial^2 \Phi_{\eta(t)}(\eta(s) - h \cdot N(s))}{\partial N(t)^2}, \tag{207}$$

and, by substituting (206), (207) into (195), obtain the identity

$$K^{2,0}_{\gamma,e}(\sigma)(s) - K^{2,0}_{\gamma,i}(\sigma)(s) =$$
$$= \frac{c(s)\, L}{2\,\pi} \cdot \lim_{h\to 0} \int_0^L \Delta^h_\eta(s,t)\,\sigma(t)\,dt + \lim_{h\to 0} \int_0^L \left(\Delta^h_\gamma(s,t) - \frac{c(s)\, L}{2\,\pi} \cdot \Delta^h_\eta(s,t)\right) \sigma(t)\,dt. \tag{208}$$

Substituting (160), (161) in Theorem 4.3 into (208), we get

$$K^{2,0}_{\gamma,e}(\sigma)(s) - K^{2,0}_{\gamma,i}(\sigma)(s) =$$
$$= -2\,\pi\, c(s)\sigma(s) + \lim_{h\to 0} \int_0^L \left(\Delta^h_\gamma(s,t) - \frac{c(s)\, L}{2\,\pi} \cdot \Delta^h_\eta(s,t)\right) \sigma(t)\,dt. \tag{209}$$

Due to Lemma 4.5, there exist positive real constants C_0, a, and h_0 such that for any $s \in [0, L]$

$$\left|\Delta^h_\gamma(s,t) - \frac{c(s)\, L}{2\,\pi} \cdot \Delta^h_\eta(s,t)\right| \le C_0, \tag{210}$$

for all $|t - s| < a$, $0 \le h < h_0$. For any $t \neq s$ and sufficiently small h, both $\Delta^h_\gamma(s,t)$ and $\Delta^h_\eta(s,t)$ are c^∞-functions. Therefore, there also exist positive real constants h_1, C_1 such that for any $s \in [0, L]$

$$\left|\Delta^h_\gamma(s,t) - \frac{c(s)\, L}{2\,\pi} \cdot \Delta^h_\eta(s,t)\right| \le C_1, \tag{211}$$

for all $|t - s| > a$, $0 \le h < h_1$. Applying Lebesgue's dominated convergence theorem (see, for example, [18]) to the second integral of the right hand side of (209), we have

$$\lim_{h\to 0} \int_0^L \left(\Delta^h_\gamma(s,t) - \frac{c(s)\, L}{2\,\pi} \cdot \Delta^h_\eta(s,t)\right) \sigma(t)\,dt = \int_0^L \lim_{h\to 0} \left(\Delta^h_\gamma(s,t) - \frac{c(s)\, L}{2\,\pi} \cdot \Delta^h_\eta(s,t)\right) \sigma(t)\,dt. \tag{212}$$

Examining (206), (207), we obviously have

$$\lim_{h\to 0} \left(\Delta^h_\gamma(s,t) - \frac{c(s)\, L}{2\,\pi} \cdot \Delta^h_\eta(s,t)\right) = 0. \tag{213}$$

Therefore, the integral on the right hand side of (212) is zero, from which (195) follows immediately. □

5 Generalizations

We have presented explicit (modulo an integral operator with a smooth kernel) formulae for integro-pseudodifferential operators of potential theory in two dimensions (up to order 2). The work presented here admits several obvious extensions.

a. Formulae (89) – (107) have their counterparts for elliptic PDEs other than the Laplace equation. Indeed, for any elliptic PDE in two dimensions, the Green's formula has the form

$$G(x,y) = \phi(x,y)\cdot\log(\|x-y\|) + \psi(x,y)\,, \tag{214}$$

with ϕ, ψ a pair of smooth functions; derivations of Section 4 are almost unchanged when $\log(\|x-y\|)$ is replaced with (214). In particular, the counterparts of the formulae (89) – (99) for the Helmholtz equation (with either real or complex Helmholtz coefficient) are identical to (89) – (99); the counterparts of the formulae (100) – (107) for the Helmholtz equation do not coincide with (100) – (107) exactly; instead, they assume the form

$$\text{(a)}\quad \begin{aligned} K^{3,0}_{\gamma,\mathrm{i}}(\sigma)(s) = {} & -2\pi\big(c(s)\big)^2\sigma(s) + 4\pi k^2\sigma(s) + \pi\sigma''(s) - 2\pi c'(s)\,H(\sigma)(s) \\ & -3\pi c(s)\,H(\sigma')(s) + N_3(\sigma)(s)\,, \end{aligned} \tag{215}$$

$$\begin{aligned} K^{3,0}_{\gamma,\mathrm{e}}(\sigma)(s) = {} & 2\pi\big(c(s)\big)^2\sigma(s) - 4\pi k^2\sigma(s) - \pi\sigma''(s) - 2\pi c'(s)\,H(\sigma)(s) \\ & -3\pi c(s)\,H(\sigma')(s) + N_3(\sigma)(s)\,, \end{aligned} \tag{216}$$

$$\text{(b)}\quad \begin{aligned} K^{2,1}_{\gamma,\mathrm{i}}(\sigma)(s) = {} & -4\pi k^2\sigma(s) - \pi\sigma''(s) + \pi c'(s)\,H(\sigma)(s) + \pi c(s)\,H(\sigma')(s) \\ & +G_3(\sigma)(s)\,, \end{aligned} \tag{217}$$

$$\begin{aligned} K^{2,1}_{\gamma,\mathrm{e}}(\sigma)(s) = {} & 4\pi k^2\sigma(s) + \pi\sigma''(s) + \pi c'(s)\,H(\sigma)(s) + \pi c(s)\,H(\sigma')(s) \\ & +G_3(\sigma)(s)\,, \end{aligned} \tag{218}$$

$$\text{(c)}\quad K^{1,2}_{\gamma,\mathrm{i}}(\sigma)(s) = 4\pi k^2\sigma(s) + \pi\sigma''(s) + \pi c(s)\,H(\sigma')(s) + \widetilde{G}_3(\sigma)(s)\,, \tag{219}$$

$$K^{1,2}_{\gamma,\mathrm{e}}(\sigma)(s) = -4\pi k^2\sigma(s) - \pi\sigma''(s) + \pi c(s)\,H(\sigma')(s) + \widetilde{G}_3(\sigma)(s)\,, \tag{220}$$

$$\text{(d)}\quad \begin{aligned} K^{0,3}_{\gamma,\mathrm{i}}(\sigma)(s) = {} & 2\pi\big(c(s)\big)^2\sigma(s) - 4\pi k^2\sigma(s) - \pi\sigma''(s) - \pi c'(s)\,H(\sigma)(s) \\ & -3\pi c(s)\,H(\sigma')(s) + \widetilde{N}_3(\sigma)(s)\,, \end{aligned} \tag{221}$$

$$\begin{aligned} K^{0,3}_{\gamma,\mathrm{e}}(\sigma)(s) = {} & -2\pi\big(c(s)\big)^2\sigma(s) + 4\pi k^2\sigma(s) + \pi\sigma''(s) - \pi c'(s)\,H(\sigma)(s) \\ & -3\pi c(s)\,H(\sigma')(s) + \widetilde{N}_3(\sigma)(s)\,, \end{aligned} \tag{222}$$

where $k \in \mathbb{C}$ is the Helmholtz coefficient, and the operators N_3, G_3, $\widetilde{N}_3$, $\widetilde{G}_3 : L^2[0,L] \to L^2[0,L]$ are compact.

b. The derivation of the three-dimensional counterparts of formulae (89) – (107) is completely straightforward; such expressions have been obtained, and the paper reporting them is in preparation.

c. In certain areas of mathematical physics, one encounters integro-pseudodifferental equations whose analysis is outside the scope of this paper. An important example is the Stratton-Chew equations, to which Maxwell's equations are frequently reduced in computational electromagnetics. Another source of such problems is the scattering of elastic waves in solids. Problems of this type are currently under investigation.

References

[1] A. J. BURTON AND G. F. MILLER, *The Application of Integral Equation Methods to the Numerical Solution of Some Exterior Boundary-value Problems*, Proc. Roy. Soc. Lond. A., 323 (1971), pp. 201–210.

[2] H. CHENG, V. ROKHLIN, AND N. YARVIN, *Non-linear Optimization, Quadrature, and Interpolation*, Tech. Rep. YALEU/DCS/RR-1169, Computer Science Department, Yale University, 1998.

[3] D. COLTON AND R. KRESS, *Integral Equation Methods in Scattering Theory*, John Wiley & Sons, 1983.

[4] M. P. DO CARMO, *Differential Geometry of Curves and Surfaces*, Prentice-Hall, 1976.

[5] M. A. EPTON AND B. DEMBART, *Multipole Translation Theory for the 3-D Laplace and Helmholtz Equations*, SIAM Journal on Scientific Computing, 10 (1995), pp. 865–897.

[6] L. GREENGARD AND V. ROKHLIN, *A New Version of the Fast Multipole Method for the Laplace Equation in Three Dimensions*, Acta Numerica, (1997), pp. 229–269.

[7] J. HADAMARD, *Lectures on the Cauchy's Problem in Linear Partial Differential Equations*, Dover, 1952.

[8] S. KAPUR AND V. ROKHLIN, *High-order Corrected Trapezoidal Rules for Singular Functions*, SIAM Journal of Numerical Analysis, 34 (1997), pp. 1331–1356.

[9] Y. KATZNELSON, *An Introduction to Harmonic Analysis*, Dover, 1976.

[10] P. KOLM AND V. ROKHLIN, *Quadruple and Octuple Layer Potentials in Two Dimensions II: Numerical Techniques*, in preparation.

[11] R. KRESS, *Linear Integral Equations*, Springer, 1989.

[12] J. R. MAUTZ AND R. F. HARRINGTON, *H-field, E-field, and Combined Field Solutions for Conducting Bodies of Revolution*, AEU, 32 (1978), pp. 157–164.

[13] S. G. MIKHLIN, *Integral Equations and Their Applications to Certain Problems in Mechanics, Mathematical Physics and Technology*, Pergamon Press, 1957.

[14] A. F. PETERSON, *The "Interior Resonance" Problem Associated with Surface Integral Equations of Electromagnetics: Numerical Consequences and a Survey of Remedies*. Journal of Electromagnetic Waves and Applications, 10 (1990), pp. 293–312.

[15] V. ROKHLIN, *End-point Corrected Trapezoidal Quadrature Rules for Singular Functions*. Computers and Mathematics with Applications, 20 (1990), pp. 51–62.

[16] A. SIDI AND M. ISRAELI, *Quadrature Methods for Periodic Singular and Weakly Singular Fredholm Integral Equations*, J. Sci. Comp., 3 (1988), pp. 201–231.

[17] J. SONG AND W. C. CHEW, *The Fast Illinois Solver Code: Requirements and Scaling Properties*, IEEE Computational Science and Engineering, (1998), pp. 19–23.

[18] R. L. WHEEDEN AND A. ZYGMUND, *Measure and Integral: An Introduction to Real-Analysis*, Marcel Dekker, 1977.

[19] Y. YAN AND I. H. SLOAN, *On Integral Equations of the First Kind with Logarithmic Kernels*, J. Integral Equations Appl., 1 (1988), pp. 549–579.

[20] S. A. YANG, *Acoustic Scattering by a Hard or Soft Body Across a Wide Frequency Range by the Helmholtz Integral Equation Method*, Journal of the Acoustical Society of America, 102 (1997), pp. 2511–2520.

Well-Conditioned Boundary Integral Equations for Three-Dimensional Electromagnetic Scattering[§]

H. Contopanagos[*], B. Dembart[†], M. Epton[†], J.J. Ottusch[*],
V. Rokhlin[‡], J. Visher[*], S. Wandzura[*]

Research Report YALEU/DCS/RR-1198
June 15, 2000

YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

We introduce a new version of the combined field integral equation (CFIE) for the solution of electromagnetic scattering problems in three dimensions. Unlike the conventional CFIE, the version reported here is well-conditioned. While we use a standard magnetic field integral operator, we precondition the electric field integral operator, converting it into a second-kind integral operator; the resulting CFIE is an integral equation of the second kind that has no spurious resonances. We also report numerical results showing that the new formulation stabilizes the number of iterations needed to solve the CFIE on closed surfaces. This is in contrast to the conventional CFIE, where the number of iterations grows as the discretization is refined.

Well-Conditioned Boundary Integral Equations for Three-Dimensional Electromagnetic Scattering[§]

H. Contopanagos*, B. Dembart†, M. Epton†, J.J. Ottusch*,
V. Rokhlin‡, J. Visher*, S. Wandzura*
Research Report YALEU/DCS/RR-1198
June 15, 2000

[§]This research was supported in part by the Defense Advanced Research Projects Agency under Contract No. MDA972-95-C-0021. *HRL Laboratories, LLC, 3011 Malibu Canyon Rd. Malibu, CA 90265. †Boeing, P.O. Box 3707, Seattle, WA 98124. ‡Yale University, Department of Computer Science, New Haven, CT 06520

Approved for public release: distribution is unlimited.

Keywords: *Electromagnetic scattering, preconditioner, combined field integral equation, Radar cross section*

1 Introduction

Recent progress in the construction of "fast" methods for the solution of the boundary integral equations of scattering theory [1] has vastly increased the size of tractable problems [2, 3]; it has also increased the need for well-conditioned boundary integral formulations. There are two principal reasons for this:

- Since we have sparse decompositions of the integral operators of scattering theory, but not their inverses, we employ iterative solvers. Well-conditioned systems of equations can be solved with few iterations.
- Using a fine discretization to resolve source variations or geometric detail on a subwavelength scale results in an ill-conditioned linear equation. This is sometimes called the "low frequency" problem in computational electromagnetics.

Only second-kind integral equations (see Appendix), or objects with similar spectral behavior (such as appropriately preconditioned differential equations) can be solved with fully controlled approximation error. The correct operators are the sum of a constant (or at least well-conditioned and easily invertible) operator and a compact operator.

Boundary integral operators of scattering typically violate this requirement in one of three ways:

- The spectrum may accumulate at zero. A typical example is the first-kind integral equation for the scalar Dirichlet problem (used for $2d$ electromagnetic scattering calculations in TM polarization),
- the operator may have an unbounded spectrum, such as a pseudodifferential or hypersingular operator,
- the operator may have small eigenvalues associated with resonances, often unphysical; the latter are often referred to as "spurious resonances" (see, for example, [4]).

For electromagnetic scattering from perfectly electrically conducting (PEC) surfaces, the standard boundary integral equations are the electric field integral equation (EFIE)

$$-\mathbf{n} \times \mathbf{E}^i = T\mathbf{J} \tag{1}$$

and the magnetic field integral equation (MFIE)

$$Z\mathbf{n} \times \mathbf{H}^i = \left(\frac{1}{2} + K\right)\mathbf{J}, \tag{2}$$

where the integral operators T and K are defined (as in [5]) by[*]

$$\begin{aligned} T\mathbf{J} &\equiv T(k) \\ &\equiv ik\mathbf{n}(\mathbf{x}) \times \int_S ds' \left\{ G(k,\mathbf{x},\mathbf{x}')\,\mathbf{J}(\mathbf{x}) + \frac{1}{k^2}\nabla\left[\nabla G(k,\mathbf{x},\mathbf{x}') \cdot \mathbf{J}(\mathbf{x}')\right] \right\} \quad (3) \\ K\mathbf{J} &\equiv K(k)\mathbf{J} \equiv -\mathbf{n}(\mathbf{x}) \times \int_S ds'\, \nabla G(k,\mathbf{x},\mathbf{x}') \times \mathbf{J}(\mathbf{x}')\,, \quad (4) \end{aligned}$$

where ∇ denotes differentiation with respect to $\mathbf{x}$, and $\mathbf{n}(\mathbf{x})$ is the unit normal to the surface at $\mathbf{x}$.

The MFIE is a second-kind integral equation. Unfortunately, this equation is suitable for an unacceptably small class of electromagnetic problems. It is inapplicable to open surfaces, becomes ill-conditioned in the presence of geometric singularities, and suffers from spurious resonances. The EFIE has both a compact piece and a hypersingular piece (coming from the double gradient term). One can eliminate the spurious resonances of the MFIE by adding the EFIE to form a combined field integral equation (CFIE) [6]. The cost of doing so is the introduction of the EFIE's hypersingular piece, which spoils the conditioning for fine discretizations (or low frequencies).

Adams and Brown [7, 8] and Kolm and Rokhlin [9] recently observed that a hypersingular integral operator and a first-kind integral operator are ideal preconditioners for each other, in the sense that the composition of the two has the spectral characteristics of a second-kind integral operator. In this letter, we show how the same approach can be employed to analytically precondition the EFIE. In fact (as was implicit in a result of Hsiao and Kleinman [5]), the electric field integral operator T preconditions itself.

Two issues raised in [8] are important for the successful application of this idea to closed bodies. First, only the local (or short distance) behavior of the preconditioner is important for asymptotic conditioning. Thus, one can precondition the EFIE by multiplying it by an electric field integral operator corresponding to an arbitrary wavenumber, real or complex; if the wavenumber has a positive imaginary part, one avoids the introduction of any additional resonances. (Obviously, if the EFIE preconditioner reproduced the MFIE resonances, then the CFIE would also have them.) Second, one must take care that the discretization of the product of preconditioner and preconditioned operators preserves the correct spectral properties.

In this letter we describe well-conditioned formulations for both open and closed surfaces. We also present numerical results for closed surfaces which demonstrate the advantages of the new CFIE formulation over the conventional CFIE.

---

[*] The other terms follow the usual conventions: $\mathbf{J} \equiv Z\mathbf{n} \times \mathbf{H}$ is the unknown surface current, $\mathbf{E}^i$ and $\mathbf{H}^i$ are the incident electric and magnetic fields, respectively, $Z = \sqrt{\mu/\epsilon}$ is the wave impedance, and $G(k,\mathbf{x},\mathbf{x}') = \exp(ikr)/4\pi r$ is the $3d$ Helmholtz kernel with $r = |\mathbf{x} - \mathbf{x}'|$ being the distance separating field and source points. Harmonic time dependence $e^{-i\omega t}$ is assumed.

2 Preconditioning the EFIE operator

References [8] and [9] consider integral operators constructed from the kernel for the Laplace and Helmholtz equations in $2d$. They observe that the product of a first-kind operator, constructed from an undifferentiated kernel, and a hypersingular operator, constructed from a twice differentiated kernel, has the desirable spectral characteristics of a second-kind operator. Since the EFIE integral operator T has both of these, one might expect that the composition of two such operators $T^2 \equiv T \circ T$ would include a constant operator and a compact operator. One might also worry that the product of hypersingular components would produce another hypersingular operator. It is easy to see, however, that the rotation operation $\mathbf{n}\times$ in the definition (3) of T, which annihilates the component of the surface vector field normal to the surface, also ensures that the product of the two hypersingular operators is identically equal to zero. Indeed, applying the hypersingular component of the second T operator to an arbitrary tangential surface vector function $\mathbf{f}(\mathbf{x}')$ produces a surface gradient function

$$\mathbf{n} \times \nabla\phi(\mathbf{x}) = \frac{i}{k}(\mathbf{n} \times \nabla) \int_S ds' \, \nabla G(k, \mathbf{x}, \mathbf{x}') \cdot \mathbf{f}(\mathbf{x}') \; , \tag{5}$$

which the hypersingular component of the first T operator, in turn, annihilates (for closed surfaces) by virtue of the identity

$$\nabla_S \cdot [\mathbf{n} \times \nabla\phi(\mathbf{x})] = 0, \tag{6}$$

with ∇_S denoting the surface gradient operator on S; identity (6), the surface analog of the $3d$ identity $\nabla \cdot [\nabla \times \phi(\mathbf{x})] = 0$, can be found, for example, in [10], and is valid for any sufficiently smooth function ϕ on S. It follows immediately from (3), (5), and (6) that T^2 behaves as a second-kind integral operator.

In this letter we investigate in detail the spectral properties of the EFIE and MFIE integral operators and combinations thereof for the PEC sphere, a simple $3d$ target for which the spectral properties of these operators are known analytically. A complete set of basis functions on the surface of a sphere of radius a is given by the vector spherical harmonics [11]

$$\mathbf{X}_{lm}(\theta, \varphi) \equiv \frac{a}{i\sqrt{l(l+1)}} \mathbf{n} \times \nabla Y_{lm}(\theta, \varphi) \, , \tag{7}$$

$$\mathbf{U}_{lm}(\theta, \varphi) \equiv \mathbf{n} \times \mathbf{X}_{lm}(\theta, \varphi) \, , \tag{8}$$

defined here in terms of the scalar spherical harmonics $Y_{lm}(\theta, \varphi)$.

The result of applying T and $K^+ \equiv \left(K + \frac{1}{2}\right)$ to each basis function is[†] [5]

$$T(k) \left\{ \begin{array}{c} \mathbf{X}_{lm} \\ \mathbf{U}_{lm} \end{array} \right\} = \left\{ \begin{array}{c} -\mathbb{J}_l(ka)\, \mathbb{H}_l(ka)\, \mathbf{U}_{lm} \\ \mathbb{J}'_l(ka)\, \mathbb{H}'_l(ka)\, \mathbf{X}_{lm} \end{array} \right\} \tag{9}$$

[†] The MFIE eigenvalues in [5] contain a sign error which is corrected here.

and

$$K^{+}(k)\left\{\begin{array}{c}\mathbf{X}_{lm}\\ \mathbf{U}_{lm}\end{array}\right\}=\left\{\begin{array}{c}i\mathbb{J}_l'(ka)\,\mathbb{H}_l(ka)\,\mathbf{X}_{lm}\\ -i\mathbb{J}_l(ka)\,\mathbb{H}_l'(ka)\,\mathbf{U}_{lm}\end{array}\right\}, \tag{10}$$

where $\mathbb{J}_l$ and $\mathbb{H}_l$ are Riccati-Bessel and (first-kind) Riccati-Hankel functions of order l, and k is the wavenumber associated with the kernel of each integral operator. The Riccati-Bessel and Riccati-Hankel functions are defined [12] in terms of spherical Bessel and Hankel functions $j_l(x)$ and $h_l^{(1)}(x)$ by

$$\mathbb{J}_l(x)\equiv xj_l(x), \tag{11}$$

$$\mathbb{H}_l(x)\equiv xh_l^{(1)}(x). \tag{12}$$

Although our chosen basis functions $\mathbf{X}_{lm}$ and $\mathbf{U}_{lm}$ are not eigenfunctions of the operator $T(k)$, they are eigenfunctions of $T^2(k)\equiv T(k)\circ T(k)$:

$$T^2(k)\left\{\begin{array}{c}\mathbf{X}_{lm}\\ \mathbf{U}_{lm}\end{array}\right\}=-\mathbb{J}_l(ka)\,\mathbb{H}_l(ka)\,\mathbb{J}_l'(ka)\,\mathbb{H}_l'(ka)\left\{\begin{array}{c}\mathbf{X}_{lm}\\ \mathbf{U}_{lm}\end{array}\right\}. \tag{13}$$

The operator $T^2(k)$ has a bounded spectrum, since, in the limit of large l, its eigenvalues accumulate at $-\frac{1}{4}$ (a result which follows from the asymptotic properties of j_l and $h_l^{(1)}$ given, for example, in [12]). However, as is evident from (10) and (13), the operator $T^2(k)$ also shares resonances (at the zeros of $\mathbb{J}_l'(ka)$ for the $\mathbf{X}_{lm}$ modes, and at the zeros of $\mathbb{J}_l(ka)$ for the $\mathbf{U}_{lm}$ modes) with the MFIE operator $K^{+}(k)$. This fact is also evident from the identity

$$T^2(k)=K^2(k)-\frac{1}{4}=K^{-}(k)\circ K^{+}(k), \tag{14}$$

(where $K^{-}\equiv K-\frac{1}{2}$) derived in [5]. Therefore, although $T^2(k)$ is a second-kind integral operator, it is not a suitable component of a resonance-free combined field integral equation for closed bodies.

As stated earlier, boundedness of the spectrum of the product of two EFIE operators (of the form (3)) is assured if they have the same short-distance behavior, a condition that does not require the two operators to share the same wavenumber (propagation constant). If we choose EFIE operators with different wavenumbers, $T(k_1)$ and $T(k_2)$, we can simultaneously obtain a bounded product and avoid MFIE resonances.

The following analysis indicates that ik is a particularly good choice for the wavenumber in the preconditioning operator (assuming that the wavenumber k is real). The eigensystem for $T(ik)\circ T(k)$ on a sphere is

$$T(ik)\circ T(k)\left\{\begin{array}{c}\mathbf{X}_{lm}\\ \mathbf{U}_{lm}\end{array}\right\}=-\left\{\begin{array}{c}\mathbb{J}_l'(ika)\,\mathbb{H}_l'(ika)\,\mathbb{J}_l(ka)\,\mathbb{H}_l(ka)\,\mathbf{X}_{lm}\\ \mathbb{J}_l(ika)\,\mathbb{H}_l(ika)\,\mathbb{J}_l'(ka)\,\mathbb{H}_l'(ka)\,\mathbf{U}_{lm}\end{array}\right\}. \tag{15}$$

It is straightforward to show (given the properties [12] of j_l and $h_l^{(1)}$) that the eigenvalues of $T(ik)\circ T(k)$ accumulate at $\frac{i}{4}$ and $-\frac{i}{4}$ for the $\mathbf{X}_{lm}$ and $\mathbf{U}_{lm}$

eigenmodes, respectively, and that $T\,(ik) \circ T\,(k)$ does not share any resonances with the MFIE operator $K^{+}(k)$.

Since $T\,(ik) \circ T\,(k)$ is a second-kind integral operator (in the sense described in the Appendix) and does not share any resonances with $K^{+}(k)$, we are finally in a position to write a well-conditioned CFIE operator. The simplest form of such an operator is

$$T\,(ik) \circ T\,(k) + \alpha K^{+}(k)\,, \tag{16}$$

where α is a constant to be chosen. In creating this CFIE operator we have preconditioned the EFIE part before adding to it the MFIE part (which is already a second-kind integral operator). The same applies to the excitation side of the equation. The resulting CFIE is

$$-T\,(ik)\left(\mathbf{n} \times \mathbf{E}^{i}\right) + \alpha Z \mathbf{n} \times \mathbf{H}^{i} = \left[T\,(ik) \circ T\,(k) + \alpha K^{+}(k)\right]\mathbf{J}\,. \tag{17}$$

The eigensystem for the CFIE operator (16) is

$$\begin{aligned} &\left[T\,(ik) \circ T\,(k) + \alpha K^{+}(k)\right]\left\{ \begin{array}{c} \mathbf{X}_{lm} \\ \mathbf{U}_{lm} \end{array} \right\} \\ &= -\left\{ \begin{array}{c} \left[\mathbb{J}_l'\,(ika)\,\mathbb{H}_l'\,(ika)\,\mathbb{J}_l\,(ka) - i\alpha\mathbb{J}_l'\,(ka)\right]\mathbb{H}_l\,(ka)\,\mathbf{X}_{lm} \\ \left[\mathbb{J}_l\,(ika)\,\mathbb{H}_l\,(ika)\,\mathbb{J}_l'\,(ka) + i\alpha\mathbb{J}_l\,(ka)\right]\mathbb{H}_l'\,(ka)\,\mathbf{U}_{lm} \end{array} \right\}. \end{aligned} \tag{18}$$

If one chooses $\alpha = \pm 1$ then, as a function of the argument ka, these eigenvalues have no zeros. For $\alpha = +1$, they circle the origin of the complex plane.

Other well-conditioned CFIE operators can be devised, for example, by preconditioning the MFIE part before combining it with the preconditioned EFIE part. We have investigated two forms:

$$T\,(ik) \circ T\,(k) + \alpha K^{+}(ik) \circ K^{+}(k) \tag{19}$$

and

$$T\,(ik) \circ T\,(k) + \alpha \mathbf{n} \times K^{+}(ik) \circ \mathbf{n} \times K^{+}(k). \tag{20}$$

Our experience shows the numerical behavior of all three CFIE formulations to be similar.

We have proven the CFIE operators in (16), (19) and (20) to be second-kind and resonance-free for spheres. However, given that the asymptotic behavior of the eigenvalues on a smooth surface stems from the short distance behavior of the kernel, we argue (following the theorems proved in [9]) that the asymptotic behavior of the various operators on spheres should also obtain for any closed surface that can be obtained by smooth deformation of a sphere. The numerical results presented in Section 4 support this argument. We also present results for a cube, which, like many targets of practical interest, has geometric singularities. These results suggest that the new CFIE formulations should be well conditioned for a wide class of closed surfaces.

3 A Different Form of the Preconditioned EFIE Operator

There are several ways to produce a Nyström discretization of the product operator $T(k_1) \circ T(k_2)$. The simplest and most straightforward approach, multiplying the discretized representations of the individual operators, can lead to numerical difficulties. The reason is that it is relatively difficult to make the discretized representations of the hypersingular part of each operator sufficiently accurate (especially for high-spatial-frequency eigenmodes) to numerically effect the cancellation that obtains analytically.

Effective discretizations of $T(k_1) \circ T(k_2)$ can be obtained either by discretizing the product operator directly or by reformulating the product operator to eliminate the product of hypersingular operators. We have not implemented the first method because of the added complexity it entails. We have implemented the second approach using a reformulated product operator that eliminates all instances of hypersingular operators. A short derivation of the reformulated equation is given below.

The first step toward obtaining a more useful form of the product operator $T(k_1) \circ T(k_2)$ is to separate each integral operator into its singular and hypersingular components. Introducing the abbreviations

$$T_1 = T(k_1), \tag{21}$$

$$T_2 = T(k_2), \tag{22}$$

we write

$$T_1 = ik_1 T_1^S + \frac{i}{k_1} T_1^H, \tag{23}$$

$$T_2 = ik_2 T_2^S + \frac{i}{k_2} T_2^H, \tag{24}$$

where

$$T_m^S \mathbf{J} \equiv \mathbf{n}(\mathbf{x}) \times \int_S ds' \; G(k_m, \mathbf{x}, \mathbf{x}') \, \mathbf{J}(\mathbf{x}'), \tag{25}$$

$$T_m^H \mathbf{J} \equiv (\mathbf{n}(\mathbf{x}) \times \nabla) \int_S ds' \; \nabla G(k_m, \mathbf{x}, \mathbf{x}') \cdot \mathbf{J}(\mathbf{x}'). \tag{26}$$

The product operator $T_1 \circ T_2$ can be expanded into four terms. Each of the two cross terms, $T_1^S \circ T_2^H$ and $T_1^H \circ T_2^S$, can be transformed (by Stokes's theorem) into the product of new, single-gradient integral operators on S plus a line integral around the boundary of S. The term formed by the product of hypersingular integral operators, $T_1^H \circ T_2^H$, reduces to a line integral. The result is further simplified by noticing that two of the three line integrals, when applied to $\mathbf{J}$, can be combined into a single term whose argument is identical to the incident electric field $\mathbf{E}^i$ by virtue of (1).

The next step is to reformulate the excitation side of the equation, taking advantage of the fact that the incident wave obeys Maxwell's equations. By applying Stokes's theorem, we rewrite the term $T_1^H\left[\mathbf{n}(\mathbf{x}') \times \mathbf{E}^i(\mathbf{x}')\right]$ as the sum of a single-gradient integral operator on $\nabla' \times \mathbf{E}^i(\mathbf{x}')$ and a line integral that exactly cancels the line integral involving $\mathbf{E}^i$ on the other side of the equation. A further simplification follows from Faraday's Law, $\nabla \times \mathbf{E} = i\omega\mu\mathbf{H}$.

The final result for the analytically preconditioned EFIE with reformulated integral operator product is

$$\begin{aligned} &-ik_1 T_1^S\left(\mathbf{n} \times \mathbf{E}^i\right) - Z\frac{k_2}{k_1}T_1^\alpha\left(\mathbf{n} \cdot \mathbf{H}^i\right) \\ &= \left(\frac{k_2}{k_1}T_1^\alpha \circ T_2^T + \frac{k_1}{k_2}T_1^\beta \circ T_2^L - k_1 k_2 T_1^S \circ T_2^S - \frac{k_1}{k_2}T_1^E \circ T_2^L\right)\mathbf{J}, \end{aligned} \tag{27}$$

where the various integral operators are defined by

$$T_m^\alpha \phi \equiv \mathbf{n}(\mathbf{x}) \times \int_S ds'\ \nabla G\left(k_m, \mathbf{x}, \mathbf{x}'\right)\phi\left(\mathbf{x}'\right), \tag{28}$$

$$T_m^\beta \phi \equiv \mathbf{n}(\mathbf{x}) \times \int_S ds'\ \mathbf{n}(\mathbf{x}') \times \nabla' G\left(k_m, \mathbf{x}, \mathbf{x}'\right)\phi\left(\mathbf{x}'\right), \tag{29}$$

$$T_m^L \mathbf{f} \equiv \int_S ds'\ \nabla G\left(k_m, \mathbf{x}, \mathbf{x}'\right) \cdot \mathbf{f}\left(\mathbf{x}'\right), \tag{30}$$

$$T_m^T \mathbf{f} \equiv \mathbf{n}(\mathbf{x}) \cdot \int_S ds'\ \nabla G\left(k_m, \mathbf{x}, \mathbf{x}'\right) \times \mathbf{f}\left(\mathbf{x}'\right), \tag{31}$$

$$T_m^S \mathbf{f} \equiv \mathbf{n}(\mathbf{x}) \times \int_S ds'\ G\left(k_m, \mathbf{x}, \mathbf{x}'\right)\mathbf{f}\left(\mathbf{x}'\right), \tag{32}$$

$$T_m^E \phi \equiv \mathbf{n}(\mathbf{x}) \times \oint_{\partial S} d\mathbf{l}'\ G\left(k_m, \mathbf{x}, \mathbf{x}'\right)\phi\left(\mathbf{x}'\right), \tag{33}$$

with $m = 1, 2$. Note that T_m^α, T_m^β, and T_m^E map scalar functions to surface vector functions, whereas T_m^L and T_m^T do the reverse. The operator on the right hand side of (27) maps surface vector functions into surface vector functions.

In the remainder of this section we discuss closed surfaces and observe that $T_1 \circ T_2$ behaves like a second-kind integral operator. For open surfaces, the situation is somewhat more complicated in that additional analytical machinery is required to convert (27) into a second-kind integral operator. We have performed such analyses for the $2d$ and $3d$ scalar cases, and will report these results in the future.

If S is a closed surface, the term $T_1^E \circ T_2^L \mathbf{J}$ vanishes, and (27) simplifies to

$$-ik_1 T_1^S\left(\mathbf{n} \times \mathbf{E}^i\right) - Z\frac{k_2}{k_1}T_1^\alpha\left(\mathbf{n} \cdot \mathbf{H}^i\right) = S\left(k_1, k_2\right)\mathbf{J}, \tag{34}$$

where

$$S_{12} \equiv S\left(k_1, k_2\right) \equiv \frac{k_2}{k_1}T_1^\alpha \circ T_2^T + \frac{k_1}{k_2}T_1^\beta \circ T_2^L - k_1 k_2 T_1^S \circ T_2^S. \tag{35}$$

We note several features of S_{12}.

First, all of the individual integral operators that comprise S_{12} involve kernels with one or no gradients on the Helmholtz Green's function G. All such integral operators are bounded.

Second, the eigenvalues of the integral operator S_{12} do not accumulate at the origin. We will demonstrate this by examining its three components $T_1^\alpha \circ T_2^T$, $T_1^\beta \circ T_2^L$, and $T_1^S \circ T_2^S$. The operator $T_1^\alpha \circ T_2^T$ is a second-kind integral operator for the transverse (divergence-free) component of $\mathbf{J}$, and is identically zero for the longitudinal (irrotational) component of $\mathbf{J}$. Likewise, the operator $T_1^\beta \circ T_2^L$ is a second-kind operator for the longitudinal component of $\mathbf{J}$, and is identically zero for the transverse component of $\mathbf{J}$. Since any surface current distribution can be decomposed into longitudinal and transverse components [11], the sum $\frac{k_2}{k_1} T_1^\alpha \circ T_2^T + \frac{k_1}{k_2} T_1^\beta \circ T_2^L$ is a second-kind integral operator; subtracting $k_1 k_2 T_1^S \circ T_2^S$, a compact operator, does not change this result. As observed in Section 2, we can avoid resonance sharing by setting $k_1 = ik$ and $k_2 = k$. In this case, the eigenvalues of S_{12} accumulate at two points, $\pm\frac{i}{4}$, rather that at $-\frac{1}{4}$.

Third, the spectrum of S_{12}, after discretization, is bounded and includes accumulation points at the expected locations. However, an accurate discretization will have zero (or very small) eigenvalues wherever the EFIE operator $T(k_2)$ has a resonance. Thus, it has to be combined with an appropriate discretization of the MFIE operator, to obtain an effective discretization of the CFIE.

Finally, it should be noted that (34) is manifestly insusceptible to the "low-frequency" problem that plagues the EFIE. Since the well-conditioned behavior of S_{12} comes from the composite operators $\frac{k_2}{k_1} T_1^\alpha \circ T_2^T$ and $\frac{k_1}{k_2} T_1^\beta \circ T_2^L$, both of whose prefactors have modulus unity (assuming $|k_1| = |k_2| = k$), and since the term $k_1 k_2 T_1^S \circ T_2^S$ tends to zero as $k \to 0$, the full operator S_{12} remains well conditioned in the limit of low frequency.

In summary, although the operators $T_1 \circ T_2$ and S_{12} have identical spectral properties for closed bodies, it is easier to construct an accurate Nyström discretization for S_{12} because it is composed of less singular integral operators. Matrix representations of S_{12} have bounded spectra, but also suffer from spurious resonances inherited from the EFIE operator $T(k_2)$. These resonances can be eliminated by combining S_{12} with $K^+(k_2)$ (or the modified MFIE operators in (19) and (20)). The result is a well-conditioned system of linear algebraic equations.

4 Numerical Results

In this section we compare the numerical performance of the conventional CFIE (referred to below as CFIE1)

$$-\mathbf{n} \times \left(\mathbf{n} \times \mathbf{E}^i\right) + Z\mathbf{n} \times \mathbf{H}^i = \left[\mathbf{n} \times T(k) + K^+(k)\right] \mathbf{J} \tag{36}$$

with the preconditioned CFIE (CFIE2)

$$\begin{aligned} &kT^S(ik)\,\mathbf{n}\times\mathbf{E}^i + iZT^\alpha(ik)\,\mathbf{n}\cdot\mathbf{H}^i - Z\mathbf{n}\times\mathbf{H}^i \\ &= \left\{i\left[-T^\alpha(ik)\circ T^T(k) + T^\beta(ik)\circ T^L(k) - k^2T^S(ik)\circ T^S(k)\right] - K^+(k)\right\}\mathbf{J} \end{aligned} \tag{37}$$

produced by combining (17) (with $\alpha = -1$) and (34) (with $k_1 = ik$ and $k_2 = k$). We discretized the individual operators in these equations using a high-order Nyström scheme [13]. In all cases, the wave impedance Z was set to unity.

We present three examples. The first example shows how the condition number of each operator, defined as the ratio of the largest to smallest singular values, depends on the fineness of discretization. Table 1 lists the condition number (CN) of the matrix representing each CFIE operator as the size of the sphere decreases. In all cases, the same discretization was used, created by placing a 6-point quadrature rule on each of the 80 nearly identical patches that cover the sphere, for a total of 960 unknowns. As the sphere radius decreases, the condition number for the CFIE2 integral operator stabilizes at about 2, whereas the condition number of the CFIE1 integral operator continues to grow in inverse proportion to the radius.

radius(λ)	CFIE1	CFIE2
1	4.2	3.04
1/4	15	2.68
1/16	59	2.04
1/64	230	1.99
1/256	940	1.97
1/1024	3800	1.97
1/4096	15000	1.97

Table 1: Condition number of CFIE matrices for shrinking PEC spheres

The second test compares iterative solver performance for the new CFIE and the conventional CFIE. The target geometry consists of two PEC spheres, one with a radius of $\lambda/2$, the other set at a resonant radius, namely, the first zero of $\mathbb{J}_1'(2\pi r/\lambda)$ or $r \approx 0.43667457\ \lambda$. The spheres are separated by a $\lambda/100$ gap. We subdivided the patches near the gap by a factor of about 10 to adequately resolve the currents, which vary rapidly there. Table 2 compares iteration counts and radar cross section (RCS) errors for several discretizations. The iterations columns list the maximum number of iterations a conjugate gradient squared (CGS) routine required to reach a residual error of 10^{-3}. A solution computed from a substantially more refined discretization provided an accuracy reference. The stated error is the root mean squared (RMS) value of the difference between the monostatic $\phi\phi$ RCS of the comparison solution and the reference solution at 181 angles. For identical discretizations, the two methods had about the same error. The data show a dramatic difference, however, in the iteration count behavior of the two methods in response to discretization refinements.

unknowns	patches	CFIE1		CFIE2	
		iterations	error	iterations	error
1496	748	60	0.46	9	0.40
4488	748	126	0.18	11	0.18
996	498	44	0.61	9	0.48
2988	498	103	0.23	12	0.16
5976	498	163	0.016	11	0.029

Table 2: Iteration count and solution error vs. discretization for two PEC spheres.

The third test also compares iterative solver performance for the two CFIE formulations. In this case the target is a cube of size 1λ. We present numerical results for five different discretizations, the first of which was derived from a mesh (i.e., a set of patches) obtained by dividing each face into four squares. The second mesh was constructed from the first one by subdividing each square into four smaller squares. The third mesh was constructed from the second by subdividing edge-touching patches in half along a line parallel to the edge; patches adjacent to two edges (i.e., corner patches) were divided into quarters. Meshes for the fourth and fifth discretizations were constructed by recursively applying the procedure by which the third mesh was constructed from the second. This process, known as patch tapering, is useful for resolving the source singularities that arise in the vicinity of geometric singularities. It also puts stress on the conventional CFIE because points near edges get close together. Table 3 lists the maximum and average number of iterations the CGS routine needed to obtain solutions for 92 independent excitations to a residual error of 10^{-3}. The total number of unknowns is the result of using a 9-point quadrature rule on each square or rectangular patch. The iteration count for CFIE2 grows very slowly with increasing taper depth, whereas for CFIE1 it increases steadily, in accordance with expectations.

unknowns	taper depth	CFIE1		CFIE2	
		max	ave	max	ave
432	0	12	6.5	10	4.3
1728	1	18	9.9	11	4.9
3888	2	26	14	11	4.9
6912	3	41	23	11	5.5
10800	4	58	36	13	5.7

Table 3: Iteration count vs. taper depth for 1λ PEC cube.

5 Conclusions and Generalizations

The classical electric field integral operator is its own perfect preconditioner, in the sense that applying it to both sides of the EFIE converts the latter into a second-kind integral equation. When the preconditioned electric field integral operator is used as a component of the CFIE, the latter is also converted into a second-kind integral equation. Furthermore, if the preconditioning electric field operator corresponds to a complex wavenumber, the resulting CFIE has no spurious resonances.

In this paper, we describe in some detail an improved CFIE for electromagnetic scattering from perfectly conducting closed surfaces, leading to a significant improvement in the performance of iterative solvers; incorporating the approach into the existing "fast" solvers is completely straightforward. The results presented here admit generalizations in several directions. The extensions discussed below are currently under investigation, and will be reported at a later date.

The approach of this paper can be applied, with minor modifications, to surface scattering with more general boundary conditions. The extension to an interface between two dielectrics, for example, is straightforward; the resulting operators have condition numbers that are in fact somewhat lower than in the case described here. While structures consisting of several dielectrics do not appear to present serious difficulties, places where several different dielectrics come in contact with each other require separate analytical treatment.

The approach of this paper has to be modified only slightly in order to obtain second kind integral equations describing electromagnetic scattering from open perfectly conducting surfaces. In this environment, the CFIE is replaced with an appropriately preconditioned EFIE, and the edge of the surface requires separate treatment. The result is a pair of coupled integral equations, one on the surface itself, and the other on the edge of the surface (which is, obviously, a curve in R^3). At this time, the theory has been constructed for the scalar case when the boundary of the surface is a sufficiently smooth curve; the analysis of open surfaces whose boundaries have corners is in progress.

A Appendix

The standard definition of a second-kind integral operator is an operator of the form

$$\lambda I + K, \tag{38}$$

where λ is a constant, I is the identity, and K is a compact operator. In scattering theory, one encounters operators of the form

$$\lambda_1 P_1 + \lambda_2 P_2 + K, \tag{39}$$

where λ_1 and λ_2 are constants and P_1 and P_2 are orthogonal projection operators such that

$$P_1 + P_2 = I. \tag{40}$$

Operators of the form (39) possess most of the desirable properties of second-kind integral operators. In a mild abuse of terminology, we refer to such expressions as second-kind integral operators throughout this letter.

References

[1] Leslie Greengard and Stephen Wandzura. Fast multipole methods. *IEEE Computational Science & Engineering*, 5(3):16–18, July 1998.

[2] John J. Ottusch, Mark A. Stalzer, John L. Visher, and Stephen M. Wandzura. Scalable electromagnetic scattering calculations for the SGI Origin 2000. In *Proceedings SC99*, Portland, OR, November 1999. IEEE.

[3] Jiming M. Song and Weng Cho Chew. The fast Illinois solver code: Requirements and scaling properties. *IEEE Computational Science & Engineering*, 5(3):19–23, July 1998.

[4] Andrew F. Peterson. The "interior resonance" problem associated with surface integral equations of electromagnetics: Numerical consequences and a survey of remedies. *Electromagnetics*, 10:293–312, 1990.

[5] George C. Hsiao and Ralph E. Kleinman. Mathematical foundations for error estimation in numerical solutions of integral equations in electromagnetics. *IEEE Transactions on Antennas and Propagation*, 45(3):316–328, March 1997.

[6] Nagayoshi Morita, Nobuaki Kumagai, and Joseph R. Mautz. *Integral Equation Methods for Electromagnetics*. Artech House, Boston, 1990.

[7] Robert J. Adams. *A Class of Robust and Efficient Iterative Methods for Wave Scattering Problems*. Ph.D. Thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA, December 1998.

[8] Robert J. Adams and Gary S. Brown. Stabilisation procedure for electric field integral equations. *Electronics Letters*, 35(23):2015–2016, November 1999.

[9] Peter Kolm and Vladimir Rokhlin. Quadruple and octuple layer potentials in two dimensions I: Analytical apparatus. Technical Report YALEU/DCS/RR-1176, Yale University, Department of Computer Science, March 1999.

[10] Barrett O'Neil. *Elementary Differential Geometry*. Academic Press, New York, 1997.

[11] John David Jackson. *Classical Electrodynamics.* John Wiley & Sons, New York, second edition, 1975.

[12] Milton Abramowitz and Irene A. Stegun. *Handbook of Mathematical Functions.* Applied Mathematics Series. National Bureau of Standards, Cambridge, 1972.

[13] Lawrence S. Canino, John J. Ottusch, Mark A. Stalzer, John L. Visher, and Stephen M. Wandzura. Numerical solution of the Helmholtz equation in 2d and 3d using a high-order Nyström discretization. *Journal of Computational Physics*, 146:627–663, 1998.

Prolate Spheroidal Wave Functions, Quadrature, and Interpolation

H. Xiao, V. Rokhlin, N. Yarvin
Research Report YALEU/DCS/RR-1199
June 27, 2000

YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

Polynomials are one of principal tools of classical numerical analysis. When a function needs to be interpolated, integrated, differentiated, etc., it is assumed to be approximated by a polynomial of a certain fixed order (though the polynomial is almost never constructed explicitly), and a treatment appropriate to such a polynomial is applied. We introduce analogous techniques based on the assumption that the function to be dealt with is band-limited, and use the well-developed apparatus of Prolate Spheroidal Wave Functions to construct quadratures, interpolation and differentiation formulae, etc. for band-limited functions. Since band-limited functions are often encountered in physics, engineering, statistics, etc. the apparatus we introduce appears to be natural in many environments. Our results are illustrated with several numerical examples.

Prolate Spheroidal Wave Functions, Quadrature, and Interpolation

H. Xiao, V. Rokhlin, N. Yarvin
Research Report YALEU/DCS/RR-1199
June 27, 2000

The authors were supported in part by DARPA/AFOSR under Contract F49620/91/C/0084
Approved for public release: distribution is unlimited.
Keywords: *Prolate Spheroidal Wave Functions, Quadrature, Interpolation, Band-Limited Functions*

Prolate Spheroidal Wave Functions, Quadrature, and Interpolation

1 Introduction

Numerical quadrature and interpolation are a well-developed part of numerical analysis; polynomials are the classical tool for the design of such schemes. Conceptually speaking, one assumes that the function is well-approximated by expressions of the form

$$\sum_{j=0}^{n} a_j x^j, \tag{1}$$

with reasonably small n, and designs algorithms that are effective for functions of the form (1) (needless to say, one almost never actually computes the coefficients $\{a_i\}$; one only uses the fact of their existence). Obviously, the polynomial approach is only effective for functions that are well-approximated by polynomials.

When one has to handle functions that are well-behaved on the whole line (for example, in signal processing), polynomials are not an appropriate tool. In such cases, trigonometric polynomials are used; existing tools are very satisfactory for dealing with functions defined and well-behaved on the whole of $\mathbb{R}^1$. Such tools, in effect, make the assumption that the functions are band-limited or nearly so; a function $f : \mathbb{R} \to \mathbb{R}$ is said to be band-limited if there exist a positive real c and a function $\sigma \in L^2[-1,1]$ such that

$$f(x) = \int_{-1}^{1} e^{icxt}\, \sigma(t)\; dt. \tag{2}$$

However, in many cases, we are confronted with band-limited functions defined on intervals (or, more generally, on compact regions in $\mathbb{R}^n$). Wave phenomena are a rich source of such functions, both in the engineering and computational contexts; they are also encountered in fluid dynamics, signal processing, and many other areas. Often, such functions can be effectively approximated by polynomials via standard tools of classical analysis. However, even when such approximations are feasible, they are usually not optimal. Smooth periodic functions are a good illustration of this observation: while they *can* be approximated by polynomials (for example, via Chebyshev or Legendre expansions), they are more efficiently approximated by Fourier expansions, both for analytical and numerical purposes. It would appear that an approach explicitly based on trigonometric polynomials could be more efficient in dealing with band-limited functions.

In the engineering context, such an apparatus was constructed more than 30 years ago (see [20]-[21], [7]-[9]). The natural tool for analyzing band-limited functions on $\mathbb{R}^1$ is the Fourier Transform, unless the functions are periodic, in which case the natural tool is the Fourier Series. The authors of [20]-[21] observe that for the analysis of band-limited functions on the interval, Prolate Spheroidal Wave Functions are likewise a natural approach. The authors also construct a multidimensional version of the theory, though their apparatus is only complete for the case of spherical regions.

The present paper constructs tools for the use of the approach of [20]-[21] in the modern computational environment. We construct a class of quadratures for band-limited functions that closely parallel the Gaussian quadratures for polynomials. The nodes are very close to being roots of appropriately chosen Prolate Spheroidal Wave Functions, the resulting quadratures are stable, and all weights are positive. As in the case of polynomials, there are interpolation, differentiation and indefinite integration schemes associated with the obtained quadratures, exact on certain classes of band-limited functions. These procedures are the main tools necessary for the numerical use of spectral discretizations based on Prolate Spheroidal Wave Functions, instead of on the usual polynomial bases. When dealing with band-limited functions, the number of nodes required by these procedures to obtain a prescribed accuracy is much less than that required by their polynomial-based counterparts. An additional bonus is the fact that the condition number of differentiation of prolate spheroidal wave functions is less than that of differentiation of the usual polynomial basis functions (see Section 8 below).

This paper is organized as follows. Section 2 summarizes various standard mathematical facts used in the remainder of the paper. Section 3 contains derivations of various results used in the algorithms described in later sections. Section 4 describes algorithms for evaluation of prolate spheroidal wave functions and associated eigenvalues. Section 5 describes a construction of quadratures for band-limited functions. Section 6 describes an alternative approach to arriving at such quadratures; it shows that roots of appropriately chosen prolate spheroidal wave functions can serve as quadrature nodes. Section 7 analyzes the use of prolate spheroidal wave functions for interpolation. Section 8 contains results of our numerical experiments with quadratures and interpolation. Section 9 contains a number of miscellaneous properties of prolate spheroidal wave functions, and Section 10 contains generalizations and conclusions.

2 Mathematical Preliminaries

As a matter of convention, in this paper the norm of a function is, unless stated otherwise, its L^2 norm:

$$||f|| = \sqrt{\int |f(x)|^2 \, dx}. \tag{3}$$

2.1 Chebyshev systems

Definition 2.1 *A sequence of functions $\phi_1, \ldots, \phi_n$ will be referred to as a* Chebyshev system *on the interval $[a, b]$ if each of them is continuous and the determinant*

$$\begin{vmatrix} \phi_1(x_1) & \cdots & \phi_1(x_n) \\ \vdots & & \vdots \\ \phi_n(x_1) & \cdots & \phi_n(x_n) \end{vmatrix} \tag{4}$$

is nonzero for any sequence of points $x_1, \ldots, x_n$ such that $a \leq x_1 < x_2 \ldots < x_n \leq b$.

An alternate definition of a Chebyshev system is that any linear combination of the functions with nonzero coefficients must have fewer than n zeros.

Examples of Chebyshev and extended Chebyshev systems include the following (additional examples can be found in [11]).

Example 2.1 *The powers $1, x, x^2, \ldots, x^n$ form an extended Chebyshev system on the interval $(-\infty, \infty)$.*

Example 2.2 *The exponentials $e^{-\lambda_1 x}, e^{-\lambda_2 x}, \ldots, e^{-\lambda_n x}$ form an extended Chebyshev system for any $\lambda_1, \ldots, \lambda_n > 0$ on the interval $[0, \infty)$.*

Example 2.3 *The functions $1, \cos x, \sin x, \cos 2x, \sin 2x, \ldots, \cos nx, \sin nx$ form a Chebyshev system on the interval $[0, 2\pi]$.*

2.2 Generalized Gaussian quadratures

A quadrature rule is an expression of the form

$$\sum_{j=1}^{n} w_j \phi(x_j), \tag{5}$$

where the points $x_j \in \mathbb{R}$ and coefficients $w_j \in \mathbb{R}$ are referred to as the nodes and weights of the quadrature, respectively. They serve as approximations to integrals of the form

$$\int_a^b \phi(x)\, \omega(x)\, dx, \tag{6}$$

with ω being an integrable non-negative function.

Quadratures are typically chosen so that the quadrature (5) is equal to the desired integral (6) for some set of functions, commonly polynomials of some fixed order. Of these, the classical Gaussian quadrature rules consist of n nodes and integrate polynomials of order $2n - 1$ exactly. In [13], the notion of a Gaussian quadrature was generalized as follows:

Definition 2.2 *A quadrature formula will be referred to as* Gaussian *with respect to a set of $2n$ functions $\phi_1, \ldots, \phi_{2n} : [a, b] \to \mathbb{R}$ and a weight function $\omega : [a, b] \to \mathbb{R}^+$, if it consists of n weights and nodes, and integrates the functions ϕ_i exactly with the weight function ω for all $i = 1, \ldots, 2n$. The weights and nodes of a Gaussian quadrature will be referred to as Gaussian weights and nodes respectively.*

The following theorem appears to be due to Markov [14, 15]; proofs of it can also be found in [12] and [11] (in a somewhat different form).

Theorem 2.1 *Suppose that the functions $\phi_1, \ldots, \phi_{2n} : [a, b] \to \mathbb{R}$ form a Chebyshev system on $[a, b]$. Suppose in addition that $\omega : [a, b] \to \mathbb{R}$ is a non-negative integrable function $[a, b] \to \mathbb{R}$. Then there exists a unique Gaussian quadrature for the functions $\phi_1, \ldots, \phi_{2n}$ on $[a, b]$ with respect to the weight function ω. The weights of this quadrature are positive.*

While the existence of Generalized Gaussian Quadratures was observed more than 100 years ago, the constructions found in [14, 15], [6, 12], [10, 11] do not easily yield numerical algorithms for the design of such quadrature formulae; such algorithms have been constructed recently (see [13, 25, 2]).

Remark 2.1 It might be worthwhile to observe here that when a Generalized Gaussian quadrature is to be constructed, the determination of its nodes tends to be the critical step (though the procedure of [13, 25, 2] determines the nodes and weights simultaneously). Indeed, once the nodes $x_1, x_2, \ldots, x_n$ have been found, the weights $w_1, w_2, \ldots, w_n$ can be determined easily as the solution of the $n \times n$ system of linear equations

$$\sum_{j=1}^{n} w_j \cdot \phi_i(x_j) = \int_a^b \phi_i(x)\, dx, \tag{7}$$

with $i = 1, 2, \ldots, n$.

2.3 Legendre Polynomials

In agreement with standard practice, we will be denoting by P_n the classical Legendre polynomials, defined by the three-term recursion

$$P_{n+1}(x) = \frac{2n+1}{n+1}\, x\, P_n(x) - \frac{n}{n+1}\, P_{n-1}(x), \tag{8}$$

with the initial conditions

$$\begin{aligned} P_0(x) &= 1, \\ P_1(x) &= x; \end{aligned} \tag{9}$$

as is well-known,

$$P_k(1) = 1 \tag{10}$$

for all $k = 0, 1, 2, \ldots$, and each of the polynomials P_k satisfies the differential equation

$$(1 - x^2)\,\frac{d^2 P_k(x)}{dx^2} - 2x\,\frac{dP_k(x)}{dx} + k \cdot (k+1)\,P_k(x) = 0. \tag{11}$$

The polynomials defined by the formulae (8),(9) are orthogonal on the interval $[-1, 1]$; however, they are not orthonormal, since for each $n \geq 0$,

$$\int_{-1}^{1} (P_n(x))^2\; dx = \frac{1}{n + 1/2}; \tag{12}$$

the normalized version of the Legendre polynomials will be denoted by $\overline{P_n}$, so that

$$\overline{P_n}(x) = P_n(x) \cdot \sqrt{n + 1/2}. \tag{13}$$

The following lemma follows immediately from the Cauchy-Schwartz inequality and from the orthogonality of the Legendre polynomials on the interval $[-1, 1]$:

Lemma 2.2 *For all integer $k \geq n$,*

$$\left| \int_{-1}^{1} x^k\, \overline{P_n}(x)\; dx \right| < \sqrt{\frac{2}{k+1}}. \tag{14}$$

For all integer $0 \leq k < n$,

$$\left| \int_{-1}^{1} x^k\, \overline{P_n}(x)\; dx \right| = 0. \tag{15}$$

2.4 Convolutional Volterra Equations

A convolutional Volterra equation of the second kind is an expression of the form

$$\varphi(x) = \int_{a}^{x} K(x - t)\,\varphi(t)\; dt \;+\; \sigma(x) \tag{16}$$

where a, b are a pair of numbers such that $a < b$, the functions $\sigma, K : [a, b] \to \mathbb{C}$ are square-integrable, and $\varphi : [a, b] \to \mathbb{C}$ is the function to be determined. Proofs of the following theorem can be found in [4], as well as in many other sources.

Theorem 2.3 *The equation (16) always has a unique solution on the interval $[a, b]$. If both functions K, σ are k times continuously differentiable, the solution φ is also k times continuously differentiable.*

2.5 Prolate Spheroidal Wave Functions

In this subsection, we summarize certain facts about the Prolate Spheroidal Wave Functions. Unless stated otherwise, all these facts can be found in [20, 17].

Given a real $c > 0$, we will denote by F_c the operator $L^2[-1,1] \to L^2[-1,1]$ defined by the formula

$$F_c(\varphi)(x) = \int_{-1}^{1} e^{icxt}\, \varphi(t)\; dt. \tag{17}$$

Obviously, F_c is compact; we will denote by $\lambda_0, \lambda_1, \ldots, \lambda_n, \ldots$ the eigenvalues of F_c ordered so that $|\lambda_{j-1}| \geq |\lambda_j|$ for all natural j. For each non-negative integer j, we will denote by ψ_j the eigenfunctions corresponding to λ_j, so that

$$\lambda_j \psi_j(x) = \int_{-1}^{1} e^{icxt}\, \psi_j(t)\; dt, \tag{18}$$

for all $x \in [-1,1]$; we adopt the convention that the functions are normalized such that $||\psi_j||_{L^2[-1,1]} = 1$, for all j.[1] The following theorem is a combination of several lemmas from [20],[6],[11].

Theorem 2.4 *For any positive real c, the eigenfunctions $\psi_0, \psi_1, \ldots,$ of the operator F_c are purely real, are orthonormal, and are complete in $L^2[-1,1]$. The even-numbered eigenfunctions are even, and the odd-numbered ones are odd. All eigenvalues of F_c are non-zero and simple; the even-numbered eigenvalues are purely real, and the odd-numbered ones are purely imaginary; in particular, $\lambda_j = i^j |\lambda_j|$. The functions ψ_i constitute a Chebychev system on the interval $[-1,1]$; in particular, the function ψ_i has exactly i zeroes on that interval, for any $i = 0, 1, \ldots,$.*

We will define the self-adjoint operator $Q_c : L^2[-1,1] \to L^2[-1,1]$ by the formula

$$Q_c(\varphi) = \frac{1}{\pi} \int_{-1}^{1} \frac{sin(c \cdot (x-t))}{x-t}\, \varphi(t)\; dt; \tag{19}$$

a simple calculation shows that

$$Q_c = \frac{c}{2\pi} \cdot F_c^* \cdot F_c, \tag{20}$$

that Q_c has the same eigenfunctions as F_c, and that the j-th (in descending order) eigenvalue μ_j of Q_c is connected with λ_j by the formula

$$\mu_j = \frac{c}{2\pi} \cdot |\lambda_j|^2. \tag{21}$$

---

[1] This convention differs from that used in [20]; however, the present paper is concerned almost exclusively with approximation of functions on $[-1,1]$, and in that context, the convention that the functions $\{\psi_j\}$ have unit norm on that interval is by far the most convenient.

The operator Q_c is obviously closely related to the operator $P_c : L^2[-\infty, \infty] \to [-\infty, \infty]$ defined by the formula

$$P_c(\varphi) = \frac{1}{\pi} \cdot \int_{-\infty}^{\infty} \frac{sin(c \cdot (x-t))}{x-t} \cdot \varphi(t)dt, \tag{22}$$

which, as is well known, is the orthogonal projection operator onto the space of functions of band limit c on $(-\infty, \infty)$.

For large c, the spectrum of Q_c consists of three parts: about $2c/\pi$ eigenvalues that are very close to 1, followed by order $\log(c)$ eigenvalues which decay exponentially from 1 to nearly 0; the remaining eigenvalues are all very close to zero. The following theorem, proven (in a slightly different form) in [19], describes the spectrum of Q_c more precisely.

Theorem 2.5 *For any positive real c and $0 < \alpha < 1$ the number N of eigenvalues of the operator Q_c that are greater than α satisfies the inequality*

$$\begin{aligned} &\frac{2c}{\pi} + \left(\frac{1}{\pi^2} \log \frac{1-\alpha}{\alpha}\right) \log(c) - 10 \cdot \log(c) < N < \\ &\frac{2c}{\pi} + \left(\frac{1}{\pi^2} \log \frac{1-\alpha}{\alpha}\right) \log(c) + 10 \cdot \log(c). \end{aligned} \tag{23}$$

By a remarkable coincidence, the eigenfunctions $\psi_0, \psi_1, \cdots, \psi_n$ of the operator Q_c turn out to be the Prolate Spheroidal Wave functions, well-known from classical Mathematical Physics (see, for example, [16]). The following theorem formalizes this statement; it is proven in a considerably more general form in [21].

Theorem 2.6 *For any $c > 0$, there exists a strictly increasing sequence of positive real numbers $\chi_0, \chi_1, \ldots$ such that for each $j \geq 0$, the differential equation*

$$(1-x^2)\,\psi''(x) - 2x\,\psi'(x) + (\chi_j - c^2\,x^2)\,\psi(x) = 0 \tag{24}$$

has a solution that is continuous on the interval $[-1, 1]$. For each $j \geq 0$, the function ψ_j (defined in Theorem 2.4) is the solution of (24).

3 Analytical Apparatus

3.1 Prolate Series

Since the functions $\psi_0, \psi_1, \ldots, \psi_n, \ldots$ are a complete orthonormal basis in $L^2[-1, 1]$, any formula for the inner product of prolate spheroidal wave functions with another function f is also a formula for the coefficients of an expansion of f into prolate spheroidal functions (which we will refer to as the prolate expansion of f). Thus the following theorem provides the coefficients of the prolate expansion of the derivative of a prolate spheroidal function, and also the coefficients of the prolate expansion of a prolate spheroidal wave function multiplied by x. Those coefficients are also the entries of the matrix for differentiation of a prolate expansion (producing another prolate expansion), and the entries of the matrix for multiplication of a prolate expansion by x, respectively. (These formulae are not, however, suitable for producing such matrices numerically, since in many cases they exhibit catastrophic cancellation.)

Theorem 3.1 *Suppose that c is real and positive, and that the integers m and n are non-negative. If $m = n \pmod 2$, then*

$$\int_{-1}^{1} \psi_n'(x)\,\psi_m(x)\;dx = \int_{-1}^{1} x\,\psi_n(x)\,\psi_m(x)\;dx = 0. \tag{25}$$

If $m \neq n \pmod 2$, then

$$\int_{-1}^{1} \psi_n'(x)\,\psi_m(x)\;dx \;=\; \frac{2\,\lambda_m^2}{\lambda_m^2+\lambda_n^2}\,\psi_m(1)\,\psi_n(1)\,, \tag{26}$$

$$\int_{-1}^{1} x\,\psi_n(x)\,\psi_m(x)\;dx \;=\; \frac{2}{ic}\,\frac{\lambda_m\lambda_n}{\lambda_m^2+\lambda_n^2}\,\psi_m(1)\,\psi_n(1)\,. \tag{27}$$

Proof. Since the functions ψ_j are alternately even and odd, (25) is obvious. In order to prove (26), we start with the identity

$$\lambda_n\psi_n = \int_{-1}^{1} e^{icxt}\,\psi_n(t)\;dt \tag{28}$$

(see (18) in Subsection 2.5). Differentiating (28) with respect to x, we obtain

$$\lambda_n\psi_n'(x) = i\,c\int_{-1}^{1} t\,e^{icxt}\,\psi_n(t)\;dt. \tag{29}$$

Projecting both sides of (29) on ψ_m and using the identity (28) (with n replaced with m) again, we have

$$\begin{aligned}
&\lambda_n\int_{-1}^{1} \psi_n'(x)\,\psi_m(x)\;dx\\
&= \;i\,c\int_{-1}^{1} \psi_m(x)\int_{-1}^{1} t\,e^{icxt}\,\psi_n(t)\;dt\,dx\\
&= \;i\,c\int_{-1}^{1} t\,\psi_n(t)\int_{-1}^{1} e^{icxt}\,\psi_m(x)\;dx\,dt\\
&= \;i\,c\,\lambda_m\int_{-1}^{1} t\,\psi_n(t)\,\psi_m(t)\;dt.
\end{aligned} \tag{30}$$

Obviously, the above calculation can be repeated with m and n exchanged, yielding the identity

$$\lambda_m \int_{-1}^{1} \psi'_m(x)\,\psi_n(x)\;dx = i\,c\,\lambda_n \int_{-1}^{1} t\,\psi_n(t)\,\psi_m(t)\;dt\,; \tag{31}$$

combining (30) with (31), we have

$$\int_{-1}^{1} \psi'_m(x)\,\psi_n(x)\;dx = \frac{\lambda_n^2}{\lambda_m^2} \int_{-1}^{1} \psi_m(x)\,\psi'_n(x)\;dx. \tag{32}$$

On the other hand, integrating the left side of (32) by parts, we have

$$\begin{aligned} &\int_{-1}^{1} \psi'_m(x)\,\psi_n(x)\;dx \\ &= \;\psi_m(1)\,\psi_n(1) \;-\; \psi_m(-1)\,\psi_n(-1) \;-\; \int_{-1}^{1} \psi'_n(x)\,\psi_m(x)\;dx. \end{aligned} \tag{33}$$

Since $m \neq n \pmod 2$, we rewrite (33) as

$$\begin{aligned} &\int_{-1}^{1} \psi'_m(x)\,\psi_n(x)\;dx \\ &= \;2\,\psi_m(1)\,\psi_n(1) - \int_{-1}^{1} \psi'_n(x)\,\psi_m(x)\;dx. \end{aligned} \tag{34}$$

Now, combining (32) and (34) and rearranging terms, we get

$$\int_{-1}^{1} \psi'_n(x)\,\psi_m(x)\;dx = \frac{2\,\lambda_m^2}{\lambda_m^2 + \lambda_n^2}\,\psi_m(1)\,\psi_n(1)\,. \tag{35}$$

Substituting (30) into (35), we get

$$\begin{aligned} &\int_{-1}^{1} x\,\psi_n(x)\,\psi_m(x)\;dx \\ &= \;\frac{1}{ic}\,\frac{\lambda_n}{\lambda_m} \int_{-1}^{1} \psi'_n(x)\,\psi_m(x)\;dx \\ &= \;\frac{1}{ic}\,\frac{\lambda_n}{\lambda_m}\,\frac{2\,\lambda_m^2}{\lambda_m^2 + \lambda_n^2}\,\psi_m(1)\,\psi_n(1) \\ &= \;\frac{2}{ic}\,\frac{\lambda_m \lambda_n}{\lambda_m^2 + \lambda_n^2}\,\psi_m(1)\,\psi_n(1). \end{aligned} \tag{36}$$

□

The following corollary, which is an immediate consequence of (32), finds use in the numerical evaluation of the eigenvalues $\{\lambda_j\}$:

Corollary 3.2 *Suppose that c is real and positive, and that the integers m and n are non-negative. If* $m \neq n \pmod 2$*, then*

$$\frac{\lambda_m^2}{\lambda_n^2} = \frac{\int_{-1}^{1} \psi_n'(x)\, \psi_m(x)\; dx}{\int_{-1}^{1} \psi_m'(x)\, \psi_n(x)\; dx}. \tag{37}$$

3.2 Decay of Legendre Coefficients of Prolate Spheroidal Wave-functions

Since each of the functions ψ_j is analytic on $\mathbb{C}$, on the interval $[-1,1]$ it can be expanded in a Legendre series of the form

$$\psi_j(x) = \sum_{k=0}^{\infty} \beta_k \overline{P_k}(x), \tag{38}$$

with the coefficients β_k decaying superalgebraically; the following two theorems establish bounds for the decay rate.

Lemma 3.3 *Let* $\overline{P_n}(x)$ *be the n-th normalized Legendre polynomial (defined in (13)). Then for any real a,*

$$\begin{aligned} &\int_{-1}^{1} e^{iax}\, \overline{P_n}(x)\, dx \\ &= \sum_{k=k_0}^{\infty} \alpha_k \int_{-1}^{1} x^{2k}\, \overline{P_n}(x)\, dx + i \sum_{k=k_0}^{\infty} \beta_k \int_{-1}^{1} x^{2k+1}\, \overline{P_n}(x)\, dx. \end{aligned} \tag{39}$$

where

$$\alpha_k = (-1)^k \frac{a^{2k}}{(2k)!}, \tag{40}$$

$$\beta_k = (-1)^k \frac{a^{2k+1}}{(2k+1)!}, \tag{41}$$

$$k_0 = \lfloor n/2 \rfloor. \tag{42}$$

Furthermore, for all integer $m \geq \lfloor e \cdot |a| \rfloor + 1$,

$$\begin{aligned} &\left| \int_{-1}^{1} e^{iax}\, \overline{P_n}(x)\; dx - \sum_{k=k_0}^{m-1} \alpha_k \int_{-1}^{1} x^{2k}\, \overline{P_n}(x)\; dx \right. \\ &\left. - i \sum_{k=k_0}^{m-1} \beta_k \int_{-1}^{1} x^{2k+1}\, \overline{P_n}(x)\; dx \right| < \left(\frac{1}{2}\right)^{2m}. \end{aligned} \tag{43}$$

In particular, if

$$n \geq 2\left(\lfloor e \cdot |a| \rfloor + 1\right), \tag{44}$$

then

$$\left| \int_{-1}^{1} e^{iax}\, \overline{P_n}(x)\; dx \right| < \left(\frac{1}{2}\right)^{n-1}. \tag{45}$$

Proof. The formula (39) follows immediately from Lemma 2.2 and Taylor's expansion of e^{iax}. In order to prove (43), we assume that m is an integer such that

$$m \geq \lfloor e \cdot |a| \rfloor + 1. \tag{46}$$

Introducing the notation

$$R_m = \sum_{k=m}^{\infty} \alpha_k \int_{-1}^{1} x^{2k}\, \overline{P_n}(x)\; dx + i \sum_{k=m}^{\infty} \beta_k \int_{-1}^{1} x^{2k+1}\, \overline{P_n}(x)\; dx, \tag{47}$$

we immediately observe that, due to Lemma 2.2 and the triangle inequality,

$$\begin{aligned} |R_m| &\leq \sum_{k=2m}^{\infty} \left(\frac{|a|^k}{k\,!} \cdot \sqrt{\frac{2}{k+1}} \right) \\ &< \sum_{k=2m}^{\infty} \frac{|a|^k}{k\,!}. \end{aligned} \tag{48}$$

Since (46) implies that

$$\frac{|a|}{2m+k} < \frac{|a|}{2m} < \frac{1}{2e} < \frac{1}{2}, \tag{49}$$

for all integer $m, k > 0$, we rewrite (48) as

$$\begin{aligned} |R_m| &< \frac{|a|^{2m}}{(2m)\,!} \cdot \left(1 + \frac{1}{2} + \frac{1}{4} + \ldots\right) \\ &< 2\,\frac{|a|^{2m}}{(2m)\,!}, \end{aligned} \tag{50}$$

and obtain (43) immediately using Stirling's formula. Finally, we obtain (45) by choosing

$$m = \lfloor e \cdot |a| \rfloor + 1. \tag{51}$$

□

Theorem 3.4 *Let $\psi_m(x)$ be the m-th prolate spheroidal function with band limit c, let $\overline{P_k}(x)$ be the k-th normalized Legendre polynomial (defined in (13)), and let λ_m be the eigenvalue which corresponds to $\psi_m(x)$ (as in Theorem 2.4). Then for all integer $m \geq 0$ and all real positive c, if*

$$k \geq 2\,(\lfloor e \cdot c \rfloor + 1)\ , \tag{52}$$

then

$$\left| \int_{-1}^{1} \psi_m(x)\, \overline{P_k}(x)\ dx \right| < \frac{1}{\lambda_m} \cdot \left(\frac{1}{2}\right)^{k-1}. \tag{53}$$

Moreover, given any $\varepsilon > 0$, if

$$k \geq 2\,(\lfloor e \cdot c \rfloor + 1) + \log_2\left(\frac{1}{\varepsilon}\right) + \log_2\left(\frac{1}{\lambda_m}\right), \tag{54}$$

then

$$\left| \int_{-1}^{1} \psi_m(x)\, \overline{P_k}(x)\ dx \right| < \varepsilon\,. \tag{55}$$

Proof. Obviously

$$\begin{aligned} & \left| \int_{-1}^{1} \psi_m(x)\, \overline{P_k}(x)\ dx \right| \\ = \ & \frac{1}{|\lambda_m|} \cdot \left| \int_{-1}^{1} \psi_m(x) \left(\int_{-1}^{1} e^{icxt}\, \overline{P_k}(t)\ dt \right)\ dx \right| \\ < \ & \frac{1}{|\lambda_m|} \int_{-1}^{1} |\psi_m(x)| \cdot \left| \int_{-1}^{1} e^{icxt}\, \overline{P_k}(t)\ dt \right| dx\,. \end{aligned} \tag{56}$$

Introducing the notation

$$a = cx, \tag{57}$$

and remembering that

$$\int_{-1}^{1} |\psi_m(x)|\, dx = 1\,, \tag{58}$$

we observe that the combination of (56), (57), (58), and Lemma 3.3 implies that

$$\begin{aligned} & \left| \int_{-1}^{1} \psi_m(x)\, P_k(x)\ dx \right| \\ < \ & \frac{1}{|\lambda_m|} \cdot \left(\frac{1}{2}\right)^{k-1} \int_{-1}^{1} |\psi_m(x)|\, dx \\ = \ & \frac{1}{|\lambda_m|} \left(\frac{1}{2}\right)^{k-1}. \end{aligned} \tag{59}$$

Substituting (54) into (53), we immediately see (55). □

4 Numerical Evaluation of Prolate Spheroidal Wave-functions

Both the classical Bouwkamp algorithm (see, for example, [1]) for the evaluation of the functions ψ_j, and the algorithm presented in this paper for the same task, are based on the expression of those functions as a Legendre series of the form

$$\psi_j(x) = \sum_{k=0}^{\infty} \alpha_k P_k(x); \tag{60}$$

since the functions ψ_j are smooth, the coefficients α_k decay superalgebraically (with bounds for that decay being given in Theorem 3.4). Substituting (60) into (24), and using (8) and (11), we obtain the well-known three-term recursion

$$\begin{aligned} &\frac{(k+2)(k+1)}{(2k+3)(2k+5)} \cdot c^2 \cdot \alpha_{k+2} + \\ &\left(k(k+1) + \frac{2k(k+1)-1}{(2k+3)(2k-1)} \cdot c^2 - \chi_j \right) \cdot \alpha_k + \\ &\frac{k(k-1)}{(2k-3)(2k-1)} \cdot c^2 \cdot \alpha_{k-2} = 0. \end{aligned} \tag{61}$$

Combining (61) with (13), we obtain the three-term recursion

$$\begin{aligned} &\frac{(k+2)(k+1)}{(2k+3)\sqrt{(2k+5)(2k+1)}} \cdot c^2 \cdot \beta^j_{k+2} + \\ &\left(k(k+1) + \frac{2k(k+1)-1}{(2k+3)(2k-1)} \cdot c^2 - \chi_j \right) \cdot \beta^j_k + \\ &\frac{k(k-1)}{(2k-1)\sqrt{(2k-3)(2k+1)}} \cdot c^2 \cdot \beta^j_{k-2} = 0 \end{aligned} \tag{62}$$

for the coefficients $\beta^j_0, \beta^j_1, \ldots$ of the expansion

$$\psi_j(x) = \sum_{k=0}^{\infty} \beta^j_k \cdot \overline{P_k}(x); \tag{63}$$

for each $j = 0, 1, 2, \ldots$, we will denote by β^j the vector in l^2 defined by the formula

$$\beta^j = (\beta^j_0, \beta^j_1, \beta^j_2, \ldots). \tag{64}$$

The following theorem restates the recursion (62) in a slightly different form.

Theorem 4.1 *The coefficients χ_i are the eigenvalues and the vectors β^i are the corresponding eigenvectors of the operator $l^2 \to l^2$ represented by the symmetric matrix A given by the formulae*

$$A_{k,k} = k(k+1) + \frac{2k(k+1)-1}{(2k+3)(2k-1)} \cdot c^2, \tag{65}$$

$$A_{k,k+2} = \frac{(k+2)(k+1)}{(2k+3)\sqrt{(2k+1)(2k+5)}} \cdot c^2, \tag{66}$$

$$A_{k+2,k} = \frac{(k+2)(k+1)}{(2k+3)\sqrt{(2k+1)(2k+5)}} \cdot c^2, \tag{67}$$

for all $k = 0, 1, 2, \ldots$, with the remainder of the entries of the matrix being zero.

In other words, the recursion (62) can be rewritten in the form

$$(A - \chi_j \cdot I)(\beta^j) = 0, \tag{68}$$

where A is separable into two symmetric tridiagonal matrices A_{even} and A_{odd}, the first consisting of the elements of A with even-numbered rows and columns and the second consisting of the elements of A with odd-numbered rows and columns. While these two matrices are infinite, and their entries do not decay much with increasing row or column number, the eigenvectors $\{\beta^j\}$ of interest (those corresponding to the first m prolate spheroidal functions) lie almost entirely in the leading rows and columns of the matrices (as shown by Theorem 3.4). Thus the evaluation of prolate spheroidal functions can be performed by the following procedure:

- 1. Generate the leading k rows and columns of A, where k is given by (54).
- 2. Split the generated portion of A into A_{even} and A_{odd}, and use a solver for the symmetric tridiagonal eigenproblem (such as that in LAPACK) to compute their eigenvectors $\{\beta^j\}$ and eigenvalues $\{\chi_j\}$.
- 3. Use the obtained values of the coefficients $\beta_0^j, \beta_1^j, \beta_2^j, \ldots$ in the expansion (63) to evaluate the function ψ_j at arbitrary points on the interval $[-1, 1]$.

Obviously steps 1 and 2 can be performed as a precomputation, for any given value of c. As a numerical diagonalization of a positive definite tridiagonal matrix with well-separated eigenvalues, this precomputation stage is numerically robust and efficient, requiring $O(cm)$ operations to construct the Legendre expansions of the form (64) for the first m prolate spheroidal functions; each subsequent evaluation of a prolate spheroidal function takes $O(c)$ operations.

4.1 Numerical Evaluation of Eigenvalues

Although the above algorithm for the evaluation of prolate spheroidal wave functions also produces the eigenvalues $\{\chi_j\}$ of the differential operator (24), it does not produce the eigenvalues $\{\lambda_j\}$ of the integral operator F_c (defined in (17)). Some of those eigenvalues can be computed using the formula

$$\lambda_j \psi_j(x) = \int_{-1}^{1} e^{icxt}\, \psi_j(t)\; dt, \tag{69}$$

evaluating the integral on the right hand side numerically; however, that evaluation obviously has a condition number of about $1/\lambda_j$, and is thus inappropriate for computing small λ_j. A well-conditioned procedure is as follows:

- 1. Use (69) to calculate λ_0, evaluating the right hand side numerically, and with $x = 0$ (so that $\psi_0(x)$ is not small).
- 2. Use the calculated λ_0, together with Corollary 3.2, to compute the absolute values $|\lambda_j|$, for $j = 1, 2, ..., m$, computing each $|\lambda_j|$ from $|\lambda_{j-1}|$ (and again, evaluating the required integrals numerically).
- 3. Use the fact that $\lambda_j = i^j |\lambda_j|$ (see Theorem 2.4) to finish the computation.

5 Quadratures for Band-Limited Functions

Since the prolate spheroidal wave functions $\psi_0, \psi_1, \ldots, \psi_n, \ldots$ constitute a complete orthonormal basis in $L^2[-1,1]$ (see Theorem 2.4),

$$e^{icxt} = \sum_{j=0}^{\infty} \left(\int_{-1}^{1} e^{icx\tau}\, \psi_j(\tau)\; d\tau \right) \psi_j(t), \tag{70}$$

for all $x, t \in [-1, 1]$; substituting (18) into (70) yields

$$e^{icxt} = \sum_{j=0}^{\infty} \lambda_j\, \psi_j(x)\, \psi_j(t), \tag{71}$$

Thus if a quadrature integrates exactly the first n eigenfunctions, that is, if

$$\sum_{k=1}^{m} w_k \psi_j(x_k) = \int_{-1}^{1} \psi_j(x)\; dx, \tag{72}$$

for all $j = 0, 1, \ldots, n-1$, then the error of the quadrature when applied to a function $f(x) = e^{icax}$, with $a \in [-1, 1]$, is given by

$$
\begin{aligned}
&\sum_{k=1}^{m} w_k e^{icax_k} - \int_{-1}^{1} e^{icax}\, dx \\
&= \sum_{k=1}^{m} w_k \left(\sum_{j=0}^{\infty} \lambda_j \, \psi_j(a) \, \psi_j(x_k) \right) - \int_{-1}^{1} \left(\sum_{j=0}^{\infty} \lambda_j \, \psi_j(a) \, \psi_j(x) \right) \, dx \\
&= \sum_{k=1}^{m} w_k \left(\sum_{j=n}^{\infty} \lambda_j \, \psi_j(a) \, \psi_j(x_k) \right) - \int_{-1}^{1} \left(\sum_{j=n}^{\infty} \lambda_j \, \psi_j(a) \, \psi_j(x) \right) \, dx. \qquad (73)
\end{aligned}
$$

Due to the orthonormality of the functions $\{\psi_j\}$,

$$
\left\| \sum_{j=n}^{\infty} \lambda_j \, \psi_j(a) \, \psi_j(x) \right\| = \sqrt{\sum_{j=n}^{\infty} |\lambda_j|^2}. \qquad (74)
$$

From (74), it is obvious that the error of integration (73) is of roughly the same magnitude as λ_n, provided that n is in the range where the eigenvalues $\{\lambda_j\}$ are decreasing exponentially (as is the case for quadratures of any useful accuracy; see Theorem 2.5) and provided in addition that the weights $\{w_k\}$ are not large.

Now, the existence of an $n/2$-point quadrature that is exact for the first n Prolate Spheriodal Wave functions follows from the combination of Theorems 2.1, 2.4; an algorithm for the numerical evaluation of nodes and weights of such quadratures can be found in [2]. An alternative procedure for the construction of quadrature formulae for band-limited functions (leading to slightly different nodes and weights) is described in the following section; a numerical comparison of the two can be found in Section 8 below.

Remark 5.1 The above text considers only the error of integration of a single exponential. For a band-limited function $g : [-1, 1] \to \mathbb{C}$ given by the formula

$$
g(x) = \int_{-1}^{1} G(t) \, e^{icxt} \, dt, \qquad (75)
$$

for some function $G : [-1, 1] \to \mathbb{C}$, the error is obviously bounded by the formula

$$
\left| \sum_{k=1}^{m} w_k g(x_k) - \int_{-1}^{1} g(x) \, dx \right| \leq \varepsilon \cdot \|G\|, \qquad (76)
$$

where ε is the maximum error of integration (73) of a single exponential, for any $t \in [-1, 1]$. While $\|G\|$ might be much larger than $\|g\|_{[-1,1]}$ (as it is if, for instance, $g = \psi_{30 \cdot n}$), if the same equation (75) is used to extend g to the rest of the real line, then by Parseval's formula $\|G\| = \|g\|_{(-\infty,\infty)}$; that is to say, although the error of such a quadrature when applied to a band-limited function is not bounded proportional to the norm of that function on the interval of integration, it is bounded proportional to the norm of that function on the entire real line.

6 Quadrature Nodes from Roots of Prolate Functions

An alternative to the approach of the previous section is to use roots of appropriate prolate spheroidal wave functions as quadrature nodes, with the weights determined via the procedure described in Remark 2.1. The following theorems provide a basis for this; numerically (see Section 8) the resulting quadrature nodes tend to be inferior to those produced by the optimization scheme of [13, 25, 2]; however, they are useful as starting points for that scheme, or as somewhat less efficient nodes which can be computed much more quickly.

6.1 Euclid Division Algorithm for Band-Limited Functions

The following two theorems constitute a straightforward extension to band-limited functions of Euclid's division algorithm for polynomials. Their proofs are quite simple, and are provided here for completeness, since the author failed to find them in the literature.

Theorem 6.1 *Suppose that* $\sigma, \varphi : [0,1] \to \mathbb{C}$ *are a pair of* c^2-*functions such that*

$$\varphi(1) \neq 0, \tag{77}$$

c *is a positive real number, and the functions* f, p *are defined by the formulae*

$$f(x) = \int_0^1 \sigma(t)\, e^{2icxt}\, dt, \tag{78}$$

$$p(x) = \int_0^1 \varphi(t)\, e^{icxt}\, dt. \tag{79}$$

Then there exist two c^1*-functions* $\eta, \xi : [0,1] \to \mathbb{C}$ *such that*

$$f(x) = p(x)\, q(x) + r(x) \tag{80}$$

for all $x \in \mathbb{R}$, *with the functions* $q, r : [0,1] \to \mathbb{R}$ *defined by the formulae*

$$q(x) = \int_0^1 \eta(t)\, e^{icxt}\, dt, \tag{81}$$

$$r(x) = \int_0^1 \xi(t)\, e^{icxt}\, dt. \tag{82}$$

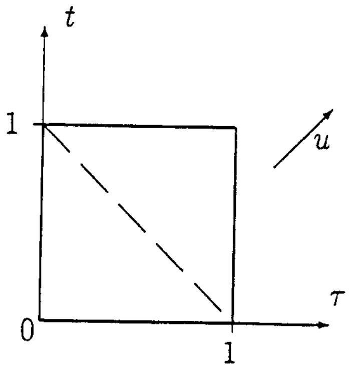


Figure 1: The split of integration range that yields (85)

Proof.

Obviously, for any functions p, q given by (79), (81),

$$
\begin{aligned}
p(x)\,q(x) &= \int_0^1 \varphi(t)\, e^{icxt}\, dt \cdot \int_0^1 \eta(\tau)\, e^{icx\tau}\, d\tau \\
&= \int_0^1 \int_0^1 \varphi(t)\, \eta(\tau)\, e^{icx(t+\tau)}\, d\tau\, dt. \qquad (83)
\end{aligned}
$$

Defining the new independent variable u by the formula

$$u = t + \tau. \qquad (84)$$

we rewrite (83) as

$$
\begin{aligned}
p(x)q(x) &= \int_0^1 e^{icux} \int_0^u \varphi(u-\tau)\, \eta(\tau)\, d\tau\, du \\
&+ \int_1^2 e^{icux} \int_{u-1}^1 \varphi(u-\tau)\, \eta(\tau)\, d\tau\, du \qquad (85)
\end{aligned}
$$

(see Figure 1). Substituting (78), (82), and (85) into (80), we get

$$
\begin{aligned}
&\int_0^1 e^{icux} \int_0^u \varphi(u-\tau)\, \eta(\tau)\, d\tau\, du \\
+ &\int_1^2 e^{icux} \int_{u-1}^1 \varphi(u-\tau)\, \eta(\tau)\, d\tau\, du + \int_0^1 \xi(t)\, e^{icxt}\, dt \\
&= \int_0^{1/2} \sigma(t)\, e^{2icxt}\, dt + \int_{1/2}^1 \sigma(t)\, e^{2icxt}\, dt. \qquad (86)
\end{aligned}
$$

Due to the well known uniqueness of the Fourier Transform, (86) is equivalent to two independent equations:

$$
\begin{aligned}
&\int_0^1 e^{icux} \int_0^u \varphi(u-\tau)\, \eta(\tau)\, d\tau\, du + \int_0^1 \xi(t)\, e^{icxt}\, dt \\
&\quad = \int_0^{1/2} \sigma(t)\, e^{2icxt}\, dt, \qquad (87)
\end{aligned}
$$

$$\int_1^2 e^{icux} \int_{u-1}^1 \varphi(u-\tau)\,\eta(\tau)\,d\tau\,du = \int_{1/2}^1 \sigma(t)\,e^{2icxt}\,dt. \tag{88}$$

Now, we observe that (88) does not contain ξ, and use it to obtain an expression for η as a function of φ, σ. After that, we will view (87) as an expression for ξ via φ, σ, η.

From (88) and the uniqueness of the Fourier Transform, we obtain

$$\int_{u-1}^1 \varphi(u-\tau)\,\eta(\tau)\,d\tau = \sigma(\frac{u}{2}), \tag{89}$$

for all $u \in [1,2]$. Introducing the new variable v via the formula

$$v = u-1, \tag{90}$$

we convert (89) into

$$\int_v^1 \varphi(v+1-\tau)\,\eta(\tau)\,d\tau = \sigma(\frac{v+1}{2}), \tag{91}$$

which is a Volterra equation of the first kind with respect to η; differentiating (91) with respect to v, we get

$$-\varphi(1)\,\eta(v) + \int_v^1 \varphi'(v+1-\tau)\,\eta(\tau)\,d\tau = \frac{1}{2}\,\sigma'(\frac{v+1}{2}), \tag{92}$$

which is a Volterra equation of the second kind. Now, the existence and uniqueness of the solution of (92) (and, therefore, of (89) and (88)) follows from Theorem 2.3 of Section 2.

With η defined as the solution of (89), we use (87) together with the uniqueness of the Fourier Transform, to finally obtain

$$\xi(u) = \sigma(\frac{u}{2}) - \int_0^u \varphi(u-\tau)\,\eta(\tau)\,d\tau, \tag{93}$$

for all $u \in [0,1]$.

□

The following theorem is a consequence of the preceding one.

Theorem 6.2 *Suppose that $\sigma, \varphi : [-1,1] \to \mathbb{C}$ are a pair of c^2–functions such that $\varphi(-1) \neq 0$, $\varphi(1) \neq 0$, c is a positive real number, and the functions f, p are defined by the formulae*

$$f(x) = \int_{-1}^1 \sigma(t)\,e^{2icxt}\,dt, \tag{94}$$

$$p(x) = \int_{-1}^1 \varphi(t)\,e^{icxt}\,dt. \tag{95}$$

Then there exist two c^1*-functions* $\eta, \xi : [-1,1] \to \mathbb{C}$ *such that*

$$f(x) = p(x)\, q(x) + r(x) \tag{96}$$

for all $x \in \mathbb{R}$*, with the functions* $q, r : [-1,1] \to \mathbb{R}$ *defined by the formulae*

$$q(x) = \int_{-1}^{1} \eta(t)\, e^{icxt}\, dt, \tag{97}$$

$$r(x) = \int_{-1}^{1} \xi(t)\, e^{icxt}\, dt. \tag{98}$$

Proof.

Defining the functions f_+, f_-,p_+, p_-, by the formulae

$$f_+(x) = \int_{0}^{1} \sigma(t)\, e^{2icxt}\, dt, \tag{99}$$

$$f_-(x) = \int_{-1}^{0} \sigma(t)\, e^{2icxt}\, dt, \tag{100}$$

$$p_+(x) = \int_{0}^{1} \varphi(t)\, e^{icxt}\, dt, \tag{101}$$

$$p_-(x) = \int_{-1}^{0} \varphi(t)\, e^{icxt}\, dt, \tag{102}$$

we observe that for all $x \in \mathbb{R}^1$,

$$f(x) = f_+(x) + f_-(x), \tag{103}$$

$$p(x) = p_+(x) + p_-(x). \tag{104}$$

Due to Theorem 6.1, there exist such η_+, η_-, ξ_+, ξ_-, that

$$f_+(x) = p_+(x)\, q_+(x) + r_+(x), \tag{105}$$

$$f_-(x) = p_-(x)\, q_-(x) + r_-(x), \tag{106}$$

with the functions q_+, q_-, r_+, r_- defined by the formulae

$$q_+(x) = \int_{0}^{1} \eta_+(t)\, e^{icxt}\, dt, \tag{107}$$

$$q_-(x) = \int_{-1}^{0} \eta_-(t)\, e^{icxt}\, dt, \tag{108}$$

$$r_+(x) = \int_0^1 \xi_+(t)\, e^{icxt}\, dt, \tag{109}$$

$$r_-(x) = \int_{-1}^0 \xi_-(t)\, e^{icxt}\, dt. \tag{110}$$

Now, defining q, by the formula

$$q(x) = q_-(x) + q_+(x) \tag{111}$$

for all $x \in [-1, 1]$, we have

$$\begin{aligned} p(x)\, q(x) &= (p_-(x) + p_+(x)) \cdot (q_-(x) + q_+(x)) \\ &= p_+(x)\, q_+(x) + p_-(x)\, q_-(x) + p_-(x)\, q_+(x) + p_+(x)\, q_-(x), \end{aligned} \tag{112}$$

and we define $r(x)$ by the obvious formula

$$r(x) = r_-(x) + r_+(x) - (p_-(x)\, q_+(x) + p_+(x)\, q_-(x)). \tag{113}$$

□

6.2 Quadrature nodes from the division theorem

In much the same way that the division theorem for polynomials can be used to provide a constructive proof of Gaussian quadratures, Theorem 6.2 provides a method of constructing generalized Gaussian quadratures for band-limited functions. The method is as follows.

To construct a quadrature for functions of a bandwidth $2c$, prolate spheroidal wave functions corresponding to bandwidth c are used. (Thus the eigenvalues $\{\lambda_j\}$ and eigenfunctions $\{\psi_j\}$ are in this section, as elsewhere in the paper, those corresponding to bandwidth c). The following theorem provides a bound of the error of a quadrature whose nodes are the roots of the n'th prolate function ψ_n, when applied to a function f which satisfies the conditions of the division theorem, in terms of the norms of the quotient and remainder of f divided by ψ_n:

Theorem 6.3 *Suppose that $x_1, x_2, \ldots, x_n \in \mathbb{R}$ are the roots of ψ_n. Let the numbers $w_1, w_2, \ldots, w_n \in \mathbb{R}$ be such that*

$$\sum_{k=1}^{n} w_k \psi_j(x_k) = \int_{-1}^{1} \psi_j(x)\, dx, \tag{114}$$

for all $j = 0, 1, \ldots, n-1$. *Then for any function* $f : [-1,1] \to \mathbb{C}$ *which satisfies the conditions of Theorem 6.2,*

$$\begin{aligned} &\left| \sum_{k=1}^{n} w_k f(x_k) - \int_{-1}^{1} f(x)\, dx \right| \\ &\quad \leq \; |\lambda_n| \cdot \|\eta\| \; + \; \|\xi\| \cdot \sum_{j=n}^{\infty} |\lambda_j| \cdot \|\psi_j\|_\infty^2 \cdot \left(2 + \sum_{k=1}^{m} \|w_k\| \right), \end{aligned} \tag{115}$$

where the functions $\eta, \xi : [-1,1] \to \mathbb{C}$ *are as defined in Theorem 6.2.*

Proof. Since f satisfies the conditions of Theorem 6.2, there exist functions $q, r : [-1,1] \to \mathbb{R}$ defined by (97),(98) such that

$$f(x) = \psi_n(x)\, q(x) + r(x). \tag{116}$$

Then, defining the error of integration E_f for the function f by

$$E_f = \left| \sum_{k=1}^{n} w_k f(x_k) - \int_{-1}^{1} f(x)\, dx \right| \tag{117}$$

we have

$$\begin{aligned} E_f \;&=\; \left| \sum_{k=1}^{n} w_k \left(\psi_n(x_k)\, q(x_k) + r(x_k) \right) - \int_{-1}^{1} (\psi_n(x)\, q(x) + r(x))\, dx \right| \\ &\leq\; \left| \sum_{k=1}^{n} w_k\, \psi_n(x_k)\, q(x_k) - \int_{-1}^{1} \psi_n(x)\, q(x)\, dx \right| \\ &\qquad + \left| \sum_{k=1}^{n} w_k r(x_k) - \int_{-1}^{1} r(x)\, dx \right| \end{aligned} \tag{118}$$

Since the nodes $\{x_k\}$ are the roots of ψ_n,

$$\sum_{k=1}^{n} w_k\, \psi_n(x_k)\, q(x_k) = 0. \tag{119}$$

Thus

$$E_f \leq \left| \int_{-1}^{1} \psi_n(x)\, q(x)\, dx \right| + \left| \sum_{k=1}^{n} w_k r(x_k) - \int_{-1}^{1} r(x)\, dx \right|. \tag{120}$$

Now

$$\begin{aligned} \int_{-1}^{1} \psi_n(x)\, q(x) dx \;&=\; \int_{-1}^{1} \psi_n(x) \int_{-1}^{1} \eta(t)\, e^{icxt}\, dt\, dx \\ &=\; \int_{-1}^{1} \eta(t) \int_{-1}^{1} \psi_n(x)\, e^{icxt}\, dx\, dt \\ &=\; \int_{-1}^{1} \eta(t)\, \lambda_n\, \psi_n(t)\, dt. \end{aligned} \tag{121}$$

Using the Cauchy-Schwartz inequality and the fact that the function ψ_n has unit norm, we get from (121) that

$$\left|\int_{-1}^{1} \psi_n(x)\, q(x)\; dx\right| \;\leq\; |\lambda_n| \cdot \|\eta\|. \tag{122}$$

Also,

$$\begin{aligned}
&\sum_{k=1}^{n} w_k r(x_k) - \int_{-1}^{1} r(x)\; dx \\
&\quad = \sum_{k=1}^{n} w_k \left(\int_{-1}^{1} \xi(t)\, e^{icx_k t}\; dt\right) - \int_{-1}^{1} \left(\int_{-1}^{1} \xi(t)\, e^{icxt}\; dt\right) dx \\
&\quad = \int_{-1}^{1} \xi(t) \left(\sum_{k=1}^{n} w_k e^{icx_k t} - \int_{-1}^{1} e^{icxt}\; dx\right) dt.
\end{aligned} \tag{123}$$

Substituting (73) into (123), and using the Cauchy-Schwartz inequality, we get

$$\begin{aligned}
&\sum_{k=1}^{n} w_k r(x_k) - \int_{-1}^{1} r(x)\; dx \\
&\quad = \int_{-1}^{1} \xi(t) \; \left(\sum_{k=1}^{m} w_k \left(\sum_{j=n}^{\infty} \lambda_j\, \psi_j(t)\, \psi_j(x_k)\right)\right. \\
&\qquad\qquad \left. - \int_{-1}^{1} \left(\sum_{j=n}^{\infty} \lambda_j\, \psi_j(t)\, \psi_j(x)\right)\; dx\right) dt \\
&\quad \leq \|\xi\| \cdot \sum_{j=n}^{\infty} |\lambda_j| \cdot \|\psi_j\|_{\infty}^{2} \cdot \left(2 + \sum_{k=1}^{m} \|w_k\|\right).
\end{aligned} \tag{124}$$

Combining (120), (122), and (124), we get

$$E_f \;\leq\; |\lambda_n| \cdot \|\eta\| \;+\; \|\xi\| \cdot \sum_{j=n}^{\infty} |\lambda_j| \cdot \|\psi_j\|_{\infty}^{2} \cdot \left(2 + \sum_{k=1}^{m} \|w_k\|\right). \tag{125}$$

□

Remark 6.1 The use of Theorem 6.3 for the construction of quadrature rules for band-limited functions depends on the fact that the norms of the band-limited functions q and r in (116) are not large, compared to the norm of f (both sets of norms being on $[-\infty, \infty]$). Such estimates have been obtained for all $n > 2c/\pi + 10\log(c)$. The proofs are quite involved, and will be reported at a later date. In this paper, we demonstrate the performance of the obtained quadrature formulae numerically (see Section 8 below).

Remark 6.2 It is natural to view (116) as an analogue for band-limited functions of the Euclid division theorem for polynomials. However, there are certain differences. In particular, Theorem 6.1 admits extensions to band-limited functions of several variables, while the classical Euclid algorithm does not. Such extensions (together with several applications) will be reported at a later date.

7 Interpolation via Prolate Spheroidal Wavefunctions

Interpolation is usually performed by the following general procedure: assuming that the function $f : [a, b] \to \mathbb{C}$ to be interpolated is given by the formula

$$f(x) = c_1\phi_1(x) + c_2\phi_2(x) + \ldots + c_n\phi_n(x), \tag{126}$$

where $\phi_1, \phi_2, \ldots, \phi_n : [a, b] \to \mathbb{C}$ are a fixed sequence of functions (often polynomials), solve an $n \times n$ linear system to determine the coefficients $c_1, c_2, \ldots, c_n$ from the values of f at the n interpolation nodes, then use (126) to evaluate f wherever needed. As is well known, if f is well-approximated by a linear combination of the interpolation functions, and if the linear system to be solved is well-conditioned, then this procedure is accurate.

As shown in Section 5 in the context of quadratures, a linear combination of the first n prolate spheroidal functions $\psi_0, \psi_1, \ldots, \psi_{n-1}$ for a band limit c can provide a good approximation to functions of the form e^{icxt}, with $t \in [-1, 1]$ (see (71,74)); in the regime where the accuracy is numerically useful, the error is of the same order of magnitude as $|\lambda_n|$. This, in turn, shows that they provide a good approximation (in the same sense as in Remark 5.1) to any band-limited function of band limit c. Thus, if $\psi_0, \psi_1, \ldots, \psi_{n-1}$ are used as the interpolation functions in this procedure, they can be expected to yield an accurate interpolation scheme for band-limited functions, provided that the matrix to be inverted is well-conditioned. The following theorem shows that if the interpolation nodes are chosen to be quadrature nodes accurate up to twice the bandwidth of interpolation, with the quadrature formula being accurate to more than twice as many digits as the interpolation formula is to be accurate to, then the matrix inverted in the procedure is close to being a scaled version of an orthogonal matrix.

Theorem 7.1 *Suppose the numbers $w_1, w_2, \ldots, w_n \in \mathbb{R}$ and $x_1, x_2, \ldots, x_n \in \mathbb{R}$ are such that*

$$\left| \int_{-1}^{1} e^{2icax}\, dx - \sum_{j=1}^{n} w_j e^{2icax_j} \right| < \varepsilon, \tag{127}$$

for all $a \in [-1,1]$, and for some $c > 0$. Let the matrix A be given by the formula

$$A = \begin{pmatrix} \psi_0(x_1) & \psi_1(x_1) & \dots & \psi_{n-1}(x_1) \\ \psi_0(x_2) & \psi_1(x_2) & \dots & \psi_{n-1}(x_2) \\ \vdots & \vdots & & \vdots \\ \psi_0(x_n) & \psi_1(x_n) & \dots & \psi_{n-1}(x_n) \end{pmatrix}, \tag{128}$$

let the matrix W be the diagonal matrix whose diagonal entries are $w_1, w_2, \dots, w_n$, and let the matrix $E = [e_{jk}]$ be given by the formula

$$E = I - A^* W A. \tag{129}$$

Then

$$|e_{jk}| < \left| \frac{2\varepsilon}{\lambda_{j-1}\lambda_{k-1}} \right|. \tag{130}$$

Proof. Clearly

$$e_{jk} = \delta_{jk} - \sum_{l=1}^{n} w_l \, \psi_{j-1}(x_l) \, \psi_{k-1}(x_l), \tag{131}$$

where δ_{ij} is the Kronecker delta function. Using (18), this becomes

$$\begin{aligned} e_{jk} &= \delta_{jk} - \sum_{l=1}^{n} w_l \cdot \left(\frac{1}{\overline{\lambda_{j-1}}} \int_{-1}^{1} e^{-icx_l t} \, \psi_{j-1}(t) \; dt \right) \\ &\qquad \cdot \left(\frac{1}{\lambda_{k-1}} \int_{-1}^{1} e^{icx_l \tau} \, \psi_{k-1}(\tau) \; d\tau \right) \\ &= \delta_{jk} - \frac{1}{\overline{\lambda_{j-1}}\lambda_{k-1}} \int_{-1}^{1} \int_{-1}^{1} \psi_{j-1}(t) \, \psi_{k-1}(\tau) \sum_{l=1}^{n} w_l e^{-icx_l t} \, e^{icx_l \tau} \; dt \, d\tau. \end{aligned} \tag{132}$$

Using (127), this becomes

$$\begin{aligned} e_{jk} = \delta_{jk} - \frac{1}{\overline{\lambda_{j-1}}\lambda_{k-1}} \int_{-1}^{1} \int_{-1}^{1} \; & \psi_{j-1}(t) \, \psi_{k-1}(\tau) \\ & \cdot \left(\int_{-1}^{1} e^{-icst} \, e^{ics\tau} \; ds \; - \; f_\varepsilon(t+\tau) \right) \; dt \, d\tau, \end{aligned} \tag{133}$$

where $f_\varepsilon : [-2,2] \to \mathbb{C}$ is a function which satisfies the relation

$$|f_\varepsilon(x)| < \varepsilon, \tag{134}$$

for all $x \in [-2,2]$. Thus

$$\begin{aligned} e_{jk} &= \delta_{jk} - \frac{1}{\overline{\lambda_{j-1}}\lambda_{k-1}} \int_{-1}^{1} \int_{-1}^{1} \psi_{j-1}(t) \, \psi_{k-1}(\tau) \int_{-1}^{1} e^{-icst} \, e^{ics\tau} \; ds \, dt \, d\tau \\ &\quad + \frac{1}{\overline{\lambda_{j-1}}\lambda_{k-1}} \int_{-1}^{1} \int_{-1}^{1} \psi_{j-1}(t) \, \psi_{k-1}(\tau) \, f_\varepsilon(t+\tau) \; dt \, d\tau \end{aligned} \tag{135}$$

Using (18), this becomes

$$\begin{aligned} e_{jk} &= \delta_{jk} - \int_{-1}^{1} \psi_{j-1}(s)\,\psi_{k-1}(s)\,ds \\ &+ \frac{1}{\lambda_{j-1}\lambda_{k-1}} \int_{-1}^{1} \psi_{k-1}(\tau) \int_{-1}^{1} \psi_{j-1}(t)\, f_\varepsilon(t+\tau)\,dt\,d\tau. \end{aligned} \tag{136}$$

Due to the orthonormality of the functions $\{\psi_j\}$, this becomes

$$e_{jk} = \frac{1}{\lambda_{j-1}\lambda_{k-1}} \int_{-1}^{1} \psi_{k-1}(\tau) \int_{-1}^{1} \psi_{j-1}(t)\, f_\varepsilon(t+\tau)\,dt\,d\tau. \tag{137}$$

Using the Cauchy-Schwartz inequality, this becomes

$$\begin{aligned} |e_{jk}| &\le \left|\frac{1}{\lambda_{j-1}\lambda_{k-1}}\right| \|\psi_{k-1}\| \sqrt{\int_{-1}^{1} \left|\int_{-1}^{1} \psi_{j-1}(t)\, f_\varepsilon(t+\tau)\,dt\right|^2 d\tau} \\ &\le \left|\frac{1}{\lambda_{j-1}\lambda_{k-1}}\right| \sqrt{\int_{-1}^{1} \|\psi_{j-1}\|^2 \int_{-1}^{1} |f_\varepsilon(t+\tau)|^2\,dt\,d\tau} \\ &= \left|\frac{1}{\lambda_{j-1}\lambda_{k-1}}\right| \sqrt{\int_{-1}^{1}\int_{-1}^{1} |f_\varepsilon(t+\tau)|^2\,dt\,d\tau} \\ &< \left|\frac{2\varepsilon}{\lambda_{j-1}\lambda_{k-1}}\right|. \end{aligned} \tag{138}$$

□

From inspection of Theorem 2.5, it can easily be seen that the number N of eigenvalues needed for a bandwidth of $2c$ and an accuracy of ε^2 is roughly twice the number of eigenvalues needed for a bandwidth of c and an accuracy of ε. Thus a generalized Gaussian quadrature for a bandwidth $2c$ and an accuracy ε^2 has roughly the same number of nodes as are needed for interpolation of accuracy ε. In our numerical experiments, this correspondence was found to be much closer than the rough bounds in Theorem 2.5 indicate; in the results tabulated in Section 8, the number of nodes for an interpolation formula of a desired accuracy ε was always chosen to be the number of quadrature nodes for a desired accuracy ε^2 for twice the band limit (that number, in turn, being chosen as indicated in Section 5); the correspondence between the desired accuracy and the experimentally measured maximum error can be seen in Tables 3 and 4.

The coefficients $c_1, c_2, \ldots, c_n$ produced by this interpolation procedure (see (126)) can, of course, just as easily be used for evaluating derivatives or indefinite integrals of the interpolated function, as they can for computing the function itself.

8 Numerical Results

The algorithms of Sections 5–7 have been implemented in double precision (64-bit floating point) arithmetic, with results shown in Tables 1–4. Tables 1 and 2 show the performance of quadrature nodes produced by the schemes of Sections 5 and 6, when used as quadrature nodes; Tables 3 and 4 show their performance when used as interpolation nodes. These are not actually the same sets of nodes; even with the bandwidth c for interpolation being half of the bandwidth for quadrature (as it is in the tables), more nodes are needed to achieve a given accuracy of interpolation than are needed to achieve a given accuracy of quadrature, as can be seen by comparing the number of nodes (printed in the column labeled n in each table). The error figures in the tables are approximations of the maximum error of interpolation or of the quadrature, when applied to functions of the form $\cos(ax)$ and $\sin(ax)$, with $0 \leq a \leq c$; they were computed by measuring the error at a large number of points in a (for interpolation, in both a and x), including the extremes. The column labeled "Roots" contains the errors for the nodes produced by the scheme of Section 6; the column labeled "Refined" contains the errors after those nodes, used as a starting point, have been run through the scheme of Section 5. The variable ε which appears in the tables is the requested accuracy, used to determine the number of nodes in the ways described in Sections 5 and 7.

Also tabulated are the numbers of Legendre nodes required to achieve the same accuracy ε using polynomial interpolation or quadrature schemes. Since Chebyshev nodes are generally known to be superior for interpolation, for that case the numbers of Chebyshev nodes required to achieve the same accuracy are also tabulated.

Figure 2 contains the maximum norm of the derivative of each prolate function $\psi_j(x)$, for $c = 200$ and $x \in [-1, 1]$, as a function of j; also graphed, for comparison, is the maximum norm of the derivative of each normalized Legendre polynomial $\overline{P}_j(x)$ over the same range; and graphed below, on the same horizontal scale, are the norms of the eigenvalues λ_j. The graph shows that, for this value of c, computing the derivatives of a function given by a prolate series is a better-conditioned operation than computing the derivatives of a function given by a Legendre series of the same number of terms. (Obviously, if the number of terms can also be reduced, as in the situations of Tables 1–4, there is a further improvement in the condition number.) The same general pattern of behavior is exhibited for other values of c; as c approaches zero (and the prolate functions approach the Legendre polynomials), the value of j at which the maximum norm of the derivative rises sharply also approaches zero (as is to be expected, since for $c = 0$ the prolate functions reduce to Legendre polynomials). Finally, Tables 5 and 6 contain samples of quadrature weights and nodes.

Remark 8.1 In this paper, detailed discussion of issues encountered in the implementation of numerical algorithms has been deliberately avoided, as well as any discussion of

CPU time requirements, memory requirements, etc. Thus, we limit ourselves to observing that all algorithms have been implemented in FORTRAN, that with the exception of the procedure for the evaluation of Prolate Spheroidal Wave functions described in Section 4, we have not designed or implemented any new or original numerical algorithms, and that the procedure of Section 4 consists of applying standard tools of numerical analysis (diagonalization of a tridiagonal matrix) to the well-known recursion (61). The resulting algorithm for the evaluation of prolate spheroidal wave functions has the CPU time requirements proportional to c^2, with a fairly large proportionality constant. The procedure of [2], when applied to the system of functions $\psi_0, \psi_1, \ldots, \psi_{2n+1}$ requires order n^3 operations, also with a fairly large proportionality constant. On the other hand, the cost of finding all roots n of the function ψ_n lying on the interval $[-1, 1]$ is proportional to n, and the proportionality constant is not large. The largest c we have dealt with in our experiments was about 6000, with resulting quadratures having about 1900 nodes. In this regime, the construction of the quadrature (both nodes and weights) took several
Mh2 minutes on the 300-megaflop SUN workstation; while there are fairly obvious ways to reduce the cost of the calculation (both in terms of asymptotic CPU time requirements and in terms of associated proportionality constants) we have made no effort to do so.

The following observations can be made from the examples presented in this section, and from the more extensive tests performed by the authors.

1. When the nodes obtained via the algorithm of [2] are used for the integration of band-limited functions, the resulting quadrature rules are significantly more accurate than the quadratures obtained from the nodes of appropriately chosen prolate functions; however, the *difference* between the numbers of nodes required by the two approaches to obtain a *prescribed* precision is not large. When the nodes obtained via the two approaches are used for the interpolation (as opposed to the integration) of band-limited functions, the performances of the two are virtually identical.

2. For large c, the number of nodes required by a quadrature rule for the integration of band-limited functions with the band-limit c is close to $\frac{c}{\pi}$; the dependence on the required precision of integration is weak (as one would expect from Theorem 2.5 and subsequent developments).

3. The numbers of nodes required by our quadratures rules to integrate band-limited functions is roughly $\pi/2$ times less than the numbers of Gaussian nodes; the numbers of nodes required by our interpolation formulae in order to interpolate band-limited functions is roughly $\pi/2$ times less than the number of Chebychev (or Gaussian) nodes. Again, the dependence of the required number of nodes on the accuracy requirements is weak.

4. The norm of the differentiation operator based on our nodes is of the order $c^{3/2}$, as

compared to the norm of the spectral differentiation operators obtained from classical polynomial expansions; this might be useful in the design of spectral (or pseudospectral) techniques.

9 Miscellaneous Properties

Prolate spheroidal wave functions possess a rich set of properties, vaguely resembling the properties of Bessel functions. This section establishes some of those properties. Some of the identities below can be found in [20],[17],[5]; others are easily derivable from the former.

The identity

$$e^{icxt} = \sum_{j=0}^{\infty} \lambda_j \, \psi_j(x) \, \psi_j(t), \tag{139}$$

(see Section 5) has a number of consequences which, while fairly obvious, seem worth recording, since similar properties of other special functions have often been found useful. Differentiating (139) m times with respect to x and n times with respect to t yields the formula

$$x^m t^n e^{icxt} = \left(\frac{1}{ic}\right)^{(m+n)} \sum_{j=0}^{\infty} \lambda_j \, \psi_j^{(m)}(x) \, \psi_j^{(n)}(t), \tag{140}$$

for all $x, t \in [-1, 1]$. Multiplying (139) by e^{-icut}, and integrating with respect to t, converts it into

$$\frac{sin(c \cdot (x-u))}{x-u} = \frac{c}{2} \sum_{j=0}^{\infty} \lambda_j^2 \, \psi_j(x) \, \psi_j(u), \tag{141}$$

Taking the squared norm of (139), and integrating with respect to x and t, yields the formula

$$\sum_{j=0}^{\infty} |\lambda_j|^2 = 4; \tag{142}$$

combining this with (21) yields

$$\sum_{j=0}^{\infty} \mu_j = \frac{2c}{\pi}. \tag{143}$$

Setting $x = t = 1$ converts (139) into

$$e^{ic} = \sum_{j=0}^{\infty} \lambda_j \psi_j^2(1). \tag{144}$$

The identity

$$\lambda_j \psi_j(x) = \int_{-1}^{1} e^{icxt} \psi_j(t)\, dt \tag{145}$$

(see Section 2.5) also has a number of simple but potentially useful consequences. Differentiating it k times with respect to x, we get

$$\lambda_j \psi_j^{(k)}(x) = (ic)^k \int_{-1}^{1} e^{icxt} t^k \psi_j(t)\, dt. \tag{146}$$

We next consider the integral

$$f(x) = f(a,x) = \int_{-1}^{1} \frac{e^{icxt}}{t-a} \psi_j(t)\, dt. \tag{147}$$

Differentiating (147) with respect to x, we have

$$\frac{d}{dx} f(a,x) = ic \int_{-1}^{1} \frac{te^{icxt}}{t-a} \psi_j(t)\, dt. \tag{148}$$

Multiplying (147) by ica, and subtracting it from (148), we obtain

$$\begin{aligned} \frac{d}{dx} f(a,x) - icaf(a,x) &= ic \int_{-1}^{1} e^{icxt} \psi_j(t)\, dt \\ &= ic\lambda_j \psi_j(x). \end{aligned} \tag{149}$$

In other words, f satisfies the differential equation

$$f'(x) - icaf(x) = ic\lambda_j \psi_j(x). \tag{150}$$

The standard "variation of parameter" calculation provides the solution to (150):

$$f(x) = ic\lambda_j \int_0^x e^{-ica(x-t)} \psi_j(t)\, dt + f(0)\, e^{icax}. \tag{151}$$

Introducing the notation

$$\mathcal{D} = \frac{1}{ic} \circ \frac{d}{dx} \tag{152}$$

(i.e. $\mathcal{D}$ is the product of multiplication by $1/ic$ and differentiation), we rewrite (146) as

$$\mathcal{D}^k(\psi_j)(x) = \frac{1}{\lambda_j}\int_{-1}^{1} t^k\, e^{icxt}\, \psi_j(t)dt; \tag{153}$$

for an arbitrary polynomial P (with real or complex coefficients),

$$P(\mathcal{D})(\psi_j)(x) = \frac{1}{\lambda_j}\int_{-1}^{1} P(t)\, e^{icxt}\, \psi_j(t)dt. \tag{154}$$

By the same token, the function ϕ defined by the formula

$$\phi(x) = \int_{-1}^{1} \frac{e^{icxt}}{P(t)}\psi_j(t)dt \tag{155}$$

satisfies the differential equation

$$P(\mathcal{D})(\phi)(x) = \lambda_m \psi_m(x). \tag{156}$$

The following lemma provides a recursion connecting the values of the k-th derivative of the function ψ_m with its derivatives of orders $k-1$, $k-2$, $k-3$, $k-4$.

Lemma 9.1 *For any positive real c, integer $m \geq 0$, and $x \in (-\infty, +\infty)$,*

$$\begin{aligned}(1-x^2)\,\psi_m^{(k+2)}(x) - 2\,(k+1)\,x\,\psi_m^{(k+1)}(x)\\ + (\chi_m - k\,(k+1) - c^2\,x^2)\,\psi_m^{(k)}(x)\\ - 2\,c^2\,k\,x\,\psi_m^{(k-1)}(x) - c^2\,k\,(k-1)\,\psi_m^{(k-2)}(x) = 0\end{aligned} \tag{157}$$

for all $k \geq 2$. Furthermore,

$$\begin{aligned}(1-x^2)\,\psi_m'''(x) - 4\,x\,\psi_m''(x) + (\chi_m - 2 - c^2\,x^2)\,\psi_m'(x)\\ - 2\,c^2\,x\,\psi_m(x) = 0\,.\end{aligned} \tag{158}$$

In particular,

$$\begin{aligned}- 2\,(k+1)\,\psi_m^{(k+1)}(1) + (\chi_m - k\,(k+1) - c^2)\,\psi_m^{(k)}(1)\\ - 2\,c^2\,k\,\psi_m^{(k-1)}(1) - c^2\,k\,(k-1)\,\psi_m^{(k-2)}(1) = 0\end{aligned} \tag{159}$$

for all $k \geq 2$, and

$$- 2\,\psi_m'(1) + (\chi_m - c^2)\,\psi_m(1) = 0\,, \tag{160}$$

$$- 4\,\psi_m''(1) + (\chi_m - 2 - c^2)\,\psi_m'(1) - 2\,c^2\,\psi_m(1) = 0\,. \tag{161}$$

Furthermore, for all integer $m \geq 0$ and $k \geq 2$,

$$\psi_m^{(k+2)}(0) + (\chi_m - k\,(k+1))\,\psi_m^{(k)}(0) \\ - c^2\,k\,(k-1)\,\psi_m^{(k-2)}(0) = 0\,. \tag{162}$$

For all odd m,

$$\psi_m'''(0) + (\chi_m - 2)\,\psi_m'(0) = 0\,, \tag{163}$$

and for all even m,

$$\psi_m''(0) + \chi_m\,\psi_m(0) = 0\,. \tag{164}$$

Finally, for all integer $m \geq 0$, $k \geq 0$,

$$\psi_m(1) \neq 0\,, \tag{165}$$

$$\psi_{2m+1}^{(2k)}(0) = 0\,, \tag{166}$$

$$\psi_{2m}^{(2k+1)}(0) = 0\,. \tag{167}$$

Proof. All of the identities (157) - (164), (166), (167), are immediately obtained by repeated differentiation of (24).
In order to prove (165), we assume that

$$\psi_m(1) = 0 \tag{168}$$

for some integer $m \geq 0$, and observe that the combination of (168) with (159), (160), (161) implies that

$$\psi_m^{(k)}(1) = 0 \tag{169}$$

for all $k = 0, 1, 2, \ldots$. Due to the analyticity of $\psi_m(x)$ in the complex plane, this would imply that

$$\psi_m(x) = 0 \tag{170}$$

for all $x \in \mathbb{R}^1$.

□

The following is an immediate consequence of the identity (160) of Lemma 9.1.

Corollary 9.2 *For all integer $m, n \geq 0$,*

$$\psi_m'(1)\cdot\psi_n(1) - \psi_n'(1)\cdot\psi_m(1) = (\chi_n - \chi_m)\cdot\psi_n(1)\cdot\psi_m(1)\,, \tag{171}$$

where $\chi_m, \chi_n \in \mathbb{R}$ are as defined in Theorem 2.6.

Theorem 3.1, in Section 3.1, gives formulae for the entries of matrices for differentiation of prolate series and for multiplication of prolate series by x. Matrices for any combination of differentiation and of multiplication by a polynomial can obviously be constructed from these two matrices; for instance, calling the differentiation matrix D, and the multiplication-by-x matrix X, the matrix for taking the second derivative of a prolate series, then multiplying it by $5 - x^2$, is equal to $(5I - X^2)D^2$.

In many cases, however, there are simpler formulae for the entries of such matrices, that is, for inner products of $\psi_j(x)$ with its derivatives and with polynomials. The following theorems establish several such formulae, as well as a few formulae for inner products which do not involve $\psi_j(x)$ itself but only its derivatives. We start with Theorem 3.1, restated here for consistency.

Theorem 9.3 *Suppose that c is real and positive, and that the integers m and n are non-negative. If $m = n \pmod 2$, then*

$$\int_{-1}^{1} \psi_n'(x)\,\psi_m(x)\;dx = \int_{-1}^{1} x\,\psi_n(x)\,\psi_m(x)\;dx = 0. \tag{172}$$

If $m \neq n \pmod 2$, then

$$\begin{aligned}\int_{-1}^{1} \psi_n'(x)\,\psi_m(x)\;dx &= \frac{2\,\lambda_m^2}{\lambda_m^2 + \lambda_n^2}\,\psi_m(1)\,\psi_n(1)\,, &(173)\\ \int_{-1}^{1} x\,\psi_n(x)\,\psi_m(x)\;dx &= \frac{2}{ic}\,\frac{\lambda_m\lambda_n}{\lambda_m^2 + \lambda_n^2}\,\psi_m(1)\,\psi_n(1)\,. &(174)\end{aligned}$$

Theorem 9.4 *Suppose that c is real and positive, and that the integers m and n are non-negative. If $m \neq n \pmod 2$, then*

$$\int_{-1}^{1} x\,\psi_n'(x)\,\psi_m(x)\;dx = 0\,. \tag{175}$$

If $m = n \pmod 2$, then

$$\int_{-1}^{1} x\,\psi_n'(x)\,\psi_m(x)\;dx = \frac{\lambda_m}{\lambda_m + \lambda_n}\,\left(2\,\psi_m(1)\,\psi_n(1) - \delta_{mn}\right)\,. \tag{176}$$

Proof. Identity (175) is obvious since the functions ψ_j are alternately even and odd (see Theorem 2.4). In order to prove (176), we consider the integral

$$\begin{aligned}&\int_{-1}^{1} x\,\psi_n'(x)\,\psi_m(x)\;dx\\ &= \frac{1}{\lambda_n}\int_{-1}^{1} x\,\left(\int_{-1}^{1} e^{icxt}\,\psi_n(t)\;dt\right)_x'\,\psi_m(x)\;dx\end{aligned}$$

$$
\begin{aligned}
&= \frac{ic}{\lambda_n}\int_{-1}^{1} x\,\psi_m(x)\,\left(\int_{-1}^{1} t\,\psi_n(t)\,e^{icxt}\,dt\right)\,dx \\
&= \frac{ic}{\lambda_n}\int_{-1}^{1} t\,\left(\int_{-1}^{1} x\,\psi_m(x)\,e^{icxt}\,dx\right)\psi_n(t)\,dt \\
&= \frac{\lambda_m}{\lambda_n}\int_{-1}^{1} t\,\psi_m'(t)\,\psi_n(t)\,dt\,.
\end{aligned}
$$

In other words,

$$
\int_{-1}^{1} x\,\psi_n'(x)\,\psi_m(x)\,dx = \frac{\lambda_m}{\lambda_n}\int_{-1}^{1} x\,\psi_m'(x)\,\psi_n(x)\,dx\,. \tag{177}
$$

On the other hand, integrating the left side of (177) by parts, we obtain

$$
\begin{aligned}
&\int_{-1}^{1} x\,\psi_n'(x)\,\psi_m(x)\,dx \\
&\quad= 2\,\psi_m(1)\,\psi_n(1) - \int_{-1}^{1} \left(\psi_n(x)\,\psi_m'(x)\,x + \psi_n(x)\,\psi_m(x)\right)\,dx \\
&\quad= 2\,\psi_m(1)\,\psi_n(1) - \int_{-1}^{1} x\,\psi_n(x)\,\psi_m'(x)\,dx - \delta_{mn}\,.
\end{aligned}
$$

Combining (177) and (178), we have

$$
\begin{aligned}
&\frac{\lambda_m}{\lambda_n}\int_{-1}^{1} x\,\psi_m'(x)\,\psi_n(x)\,dx \\
&\quad= 2\psi_m(1)\,\psi_n(1) - \int_{-1}^{1} x\,\psi_m'(x)\,\psi_n(x)\,dx - \delta_{mn}\,,
\end{aligned}
$$

from which (176) follows directly. □

Theorem 9.5 *Suppose that c is real and positive, and that the integers m and n are non-negative. If* $m \neq n \pmod 2$, *then*

$$
\int_{-1}^{1} x^2\,\psi_n''(x)\,\psi_m(x)\,dx = 0\,. \tag{178}
$$

If $m = n \pmod 2$ *and* $m \neq n$, *then*

$$
\begin{aligned}
&\int_{-1}^{1} x^2\,\psi_m''(x)\,\psi_n(x)\,dx \\
&\quad= \frac{2\lambda_n}{\lambda_m - \lambda_n}\left(\psi_n'(1)\,\psi_m(1) - \psi_m'(1)\,\psi_n(1)\right) \\
&\qquad - \frac{4\lambda_n}{\lambda_n + \lambda_m}\,\psi_n(1)\,\psi_m(1)
\end{aligned} \tag{179}
$$

$$\begin{aligned} &= \frac{\lambda_n}{\lambda_m - \lambda_n}\,(\chi_n - \chi_m)\,\psi_n(1)\,\psi_m(1) \\ &\quad - \frac{4\lambda_n}{\lambda_n + \lambda_m}\,\psi_n(1)\,\psi_m(1), \end{aligned} \tag{180}$$

where $\chi_m, \chi_n \in \mathbb{R}$ *are as defined in Theorem 2.6.*

Proof. Clearly (178) is true, since the functions ψ_j are alternately even and odd. In order to prove (179) and (180), supposing that $m = n \pmod 2$ and $m \neq n$, we consider the integral

$$\begin{aligned} &\int_{-1}^{1} x^2\,\psi_n''(x)\,\psi_m(x)\,dx \\ &= \frac{1}{\lambda_n}\int_{-1}^{1} x^2 \cdot \left(\int_{-1}^{1} e^{icxt}\,\psi_n(t)\,dt\right)''_x \psi_m(x)\,dx \\ &= -\frac{c^2}{\lambda_n}\int_{-1}^{1} \psi_m(x)\,x^2 \cdot \left(\int_{-1}^{1} t^2\,\psi_n(t)\,e^{icxt}\,dt\right)\,dx \\ &= -\frac{c^2}{\lambda_n}\int_{-1}^{1} \left(\int_{-1}^{1} \psi_m(x)\,x^2\,e^{icxt}\,dx\right)\,\psi_n(t)\,t^2\,dt \\ &= \frac{\lambda_m}{\lambda_n}\int_{-1}^{1} t^2\,\psi_n(t)\,\psi_m''(t)\,dt\,, \end{aligned}$$

which is summarized as

$$\int_{-1}^{1} x^2\,\psi_n''(x)\,\psi_m(x)\,dx = \frac{\lambda_m}{\lambda_n}\int_{-1}^{1} x^2\,\psi_m''(x)\,\psi_n(x)\,dx\,. \tag{181}$$

On the other hand, integrating the left side of (181) by parts, we have

$$\begin{aligned} &\int_{-1}^{1} x^2\,\psi_n''(x)\,\psi_m(x)\,dx \\ &= 2\,\psi_n'(1)\,\psi_m(1) - \int_{-1}^{1} \psi_n'(x)\,\left(\psi_m'(x)\,x^2 + 2\,x\,\psi_m(x)\right)\,dx \\ &= 2\,\psi_n'(1)\,\psi_m(1) - 2\int_{-1}^{1} \psi_n'(x)\,\psi_m(x)\,x\,dx \\ &\quad - \int_{-1}^{1} \psi_n'(x)\,\psi_m'(x)\,x^2\,dx\,. \end{aligned} \tag{182}$$

Due to Theorem 9.4 and the fact that $m \neq n$, we immediately rewrite (182) as

$$\begin{aligned} &\int_{-1}^{1} x^2\,\psi_n''(x)\,\psi_m(x)\,dx \\ &= 2\psi_n'(1)\,\psi_m(1) - \frac{2\lambda_m}{\lambda_m + \lambda_n}\,2\,\psi_n(1)\,\psi_m(1) \\ &\quad - \int_{-1}^{1} x^2\,\psi_n'(x)\,\psi_m'(x)\,dx\,, \end{aligned} \tag{183}$$

which we rewrite as

$$\begin{aligned}
&\int_{-1}^{1} x^2\,\psi_n'(x)\,\psi_m'(x)\;dx \\
&= \; 2\,\psi_n'(1)\,\psi_m(1) - \frac{4\,\lambda_m}{\lambda_m+\lambda_n}\,\psi_n(1)\,\psi_m(1) \\
&\quad - \int_{-1}^{1} x^2\,\psi_n''(x)\,\psi_m(x)\;dx\,. \qquad (184)
\end{aligned}$$

Swapping m with n, we convert (184) into

$$\begin{aligned}
&\int_{-1}^{1} x^2\,\psi_n'(x)\,\psi_m'(x)\;dx \\
&= \; 2\,\psi_m'(1)\,\psi_n(1) - \frac{4\,\lambda_n}{\lambda_m+\lambda_n}\,\psi_n(1)\,\psi_m(1) \\
&\quad - \int_{-1}^{1} x^2\,\psi_m''(x)\,\psi_n(x)\;dx. \qquad (185)
\end{aligned}$$

Combining (184) and (185), we obtain

$$\begin{aligned}
&\int_{-1}^{1} x^2\,\psi_n''(x)\,\psi_m(x)\;dx - 2\,\psi_n'(1)\,\psi_m(1) + \frac{4\,\lambda_m}{\lambda_m+\lambda_n}\,\psi_n(1)\,\psi_m(1) \\
= \;&\int_{-1}^{1} x^2\,\psi_m''(x)\,\psi_n(x)\;dx - 2\,\psi_m'(1)\,\psi_n(1) + \frac{4\,\lambda_n}{\lambda_m+\lambda_n}\,\psi_n(1)\,\psi_m(1)\,, \qquad (186)
\end{aligned}$$

which is obviously equivalent to

$$\begin{aligned}
&\int_{-1}^{1} x^2\,\psi_n''(x)\,\psi_m(x)\;dx \\
&= \int_{-1}^{1} x^2\,\psi_m''(x)\,\psi_n(x)\;dx + 2\,\left(\psi_n'(1)\,\psi_m(1) - \psi_m'(1)\,\psi_n(1)\right) \\
&\quad + 4\,\frac{\lambda_n-\lambda_m}{\lambda_n+\lambda_m}\,\psi_n(1)\,\psi_m(1)\,.
\end{aligned}$$

Finally, combining (181) with (187), we have

$$\begin{aligned}
&\frac{\lambda_m}{\lambda_n}\int_{-1}^{1} x^2\,\psi_m''(x)\,\psi_n(x)\;dx \\
&= \int_{-1}^{1} x^2\,\psi_m''(x)\,\psi_n(x)\;dx + 2\,\left(\psi_n'(1)\,\psi_m(1) - \psi_m'(1)\,\psi_n(1)\right) \\
&\quad + 4\,\frac{\lambda_n-\lambda_m}{\lambda_n+\lambda_m}\,\psi_n(1)\,\psi_m(1)\,, \qquad (187)
\end{aligned}$$

which is easily rewritten as

$$\begin{aligned}
&\left(\frac{\lambda_m}{\lambda_n} - 1\right) \int_{-1}^{1} x^2\, \psi_m''(x)\, \psi_n(x)\ dx \\
&\quad = \ 2\,(\psi_n'(1)\,\psi_m(1) - \psi_m'(1)\,\psi_n(1)) \\
&\qquad + 4\,\frac{\lambda_n - \lambda_m}{\lambda_n + \lambda_m}\,\psi_n(1)\,\psi_m(1)\,,
\end{aligned}$$

or

$$\begin{aligned}
&\int_{-1}^{1} x^2\, \psi_m''(x)\, \psi_n(x)\ dx \\
&\quad = \ \frac{2\,\lambda_n}{\lambda_m - \lambda_n}\,(\psi_n'(1)\,\psi_m(1) - \psi_m'(1)\,\psi_n(1)) \\
&\qquad - \frac{4\,\lambda_n}{\lambda_n + \lambda_m}\,\psi_n(1)\,\psi_m(1)\,. \qquad (188)
\end{aligned}$$

We finally rewrite (188) as (180) using Corollary 9.2. □

The following theorem is an immediate consequence of combining the preceding theorem with equation (184) from its proof.

Theorem 9.6 *Suppose that c is real and positive, and that the integers m and n are non-negative. If $m \neq n$ (mod 2), then*

$$\int_{-1}^{1} x^2\, \psi_n'(x)\, \psi_m'(x)\ dx = 0\,. \qquad (189)$$

If $m = n$ (mod 2) and $m \neq n$,

$$\begin{aligned}
&\int_{-1}^{1} x^2\, \psi_m'(x)\, \psi_n'(x)\ dx \\
&\quad = \ 2\,\psi_m'(1)\,\psi_n(1) + \frac{2\lambda_n}{\lambda_m - \lambda_n}\,(\psi_m'(1)\,\psi_n(1) - \psi_n'(1)\,\psi_m(1)) \qquad (190) \\
&\quad = \ 2\,\psi_n'(1)\,\psi_m(1) + \frac{2\lambda_m}{\lambda_n - \lambda_m}\,(\psi_n'(1)\,\psi_m(1) - \psi_m'(1)\,\psi_n(1)) \qquad (191) \\
&\quad = \ \psi_m(1)\,\psi_n(1)\left(\frac{\lambda_m\chi_m - \lambda_n\chi_n}{\lambda_m - \lambda_n} - c^2\right). \qquad (192)
\end{aligned}$$

Theorem 9.7 *Suppose that c is real and positive, and that the integers m and n are non-negative. If $m \neq n$ (mod 2), then*

$$\int_{-1}^{1} \psi_n(x)\, \psi_m''(x)\ dx = \int_{-1}^{1} x^2\, \psi_n(x)\, \psi_m(x)\ dx = 0 \qquad (193)$$

If $m = n \pmod 2$ *and* $m \neq n$, *then*

$$\int_{-1}^{1} \psi_n(x)\, \psi_m''(x)\, dx$$

$$= \frac{2\,\lambda_n^2}{\lambda_m^2 - \lambda_n^2} \left(\psi_n'(1)\,\psi_m(1) - \psi_n(1)\,\psi_m'(1)\right) \tag{194}$$

$$= \frac{\lambda_n^2}{\lambda_m^2 - \lambda_n^2} (\chi_n - \chi_m)\, \psi_m(1)\, \psi_n(1), \tag{195}$$

$$\int_{-1}^{1} x^2\, \psi_n(x)\, \psi_m(x)\, dx$$

$$= -\frac{2}{c^2} \frac{\lambda_m \lambda_n}{\lambda_m^2 - \lambda_n^2} \left(\psi_n'(1)\,\psi_m(1) - \psi_n(1)\,\psi_m'(1)\right) \tag{196}$$

$$= -\frac{1}{c^2} \frac{\lambda_m \lambda_n}{\lambda_m^2 - \lambda_n^2} (\chi_n - \chi_m)\, \psi_m(1)\, \psi_n(1), \tag{197}$$

where $\chi_m, \chi_n \in \mathbb{R}$ *are as defined in Theorem 2.6.*

Proof. Identity (193) is obvious, since the functions ψ_j are alternately even and odd. In order to prove (194)–(197), we start with the expression

$$\lambda_n \psi_n''(x) = -c^2 \int_{-1}^{1} t^2\, e^{icxt}\, \psi_n(t)\, dt\,. \tag{198}$$

Taking the inner product of (198) with $\psi_m(x)$, we have

$$\lambda_n \int_{-1}^{1} \psi_n''(x)\, \psi_m(x)\, dx$$

$$= -c^2 \int_{-1}^{1} \left(\int_{-1}^{1} t^2\, \psi_n(t)\, e^{icxt}\, dt\right) \psi_m(x)\, dx$$

$$= -c^2 \int_{-1}^{1} t^2\, \psi_n(t) \left(\int_{-1}^{1} \psi_m(x)\, e^{icxt}\, dx\right) dt$$

$$= -c^2 \lambda_m \int_{-1}^{1} t^2\, \psi_n(t)\, \psi_m(t)\, dt\,,$$

which we summarize as

$$\int_{-1}^{1} x^2\, \psi_n(x)\, \psi_m(x)\, dx = -\frac{1}{c^2} \frac{\lambda_n}{\lambda_m} \int_{-1}^{1} \psi_n''(x)\, \psi_m(x)\, dx\,. \tag{199}$$

Swapping n, m, we rewrite (199) in the form of

$$\int_{-1}^{1} x^2\, \psi_n(x)\, \psi_m(x) dx$$

$$= -\frac{1}{c^2} \frac{\lambda_m}{\lambda_n} \int_{-1}^{1} \psi_m''(x)\, \psi_n(x)\, dx\,. \tag{200}$$

Combining (199) and (200), we get

$$\int_{-1}^{1} \psi_n''(x)\,\psi_m(x)\;dx = \frac{\lambda_m^2}{\lambda_n^2}\int_{-1}^{1}\;\psi_m''(x)\,\psi_n(x)\;dx\,. \tag{201}$$

On the other hand, integrating the left side of (201) by parts, we have

$$\begin{aligned}
&\int_{-1}^{1} \psi_n''(x)\,\psi_m(x)\;dx \\
&= \;\psi_n'(1)\,\psi_m(1) - \psi_n'(-1)\,\psi_m(-1) - \int_{-1}^{1}\;\psi_n'(x)\,\psi_m'(x)\;dx \\
&= \;2\,\psi_n'(1)\,\psi_m(1) - (\psi_n(1)\,\psi_m'(1) - \psi_n(-1)\,\psi_m'(-1)) \\
&\quad + \int_{-1}^{1}\;\psi_n(x)\,\psi_m''(x)\;dx.
\end{aligned} \tag{202}$$

We rewrite (202) in the form of

$$\begin{aligned}
&\int_{-1}^{1} \psi_n''(x)\,\psi_m(x)\;dx \\
&= \;2\,(\psi_n'(1)\,\psi_m(1) - \psi_n(1)\,\psi_m'(1)) + \int_{-1}^{1}\psi_n(x)\,\psi_m''(x)\;dx\,.
\end{aligned}$$

We combine (201) and (203) and get

$$\begin{aligned}
&\left(\frac{\lambda_m^2}{\lambda_n^2} - 1\right)\int_{-1}^{1}\;\psi_n(x)\,\psi_m''(x)\;dx \\
&= \;2\;(\psi_n'(1)\,\psi_m(1) - \psi_n(1)\,\psi_m'(1))\;.
\end{aligned} \tag{203}$$

Since $m \neq n$, we easily rewrite (203) as (194). We obtain expression (196) by combining (200) and (194). The identities (195), (197) follow from (194), (196) immediately due to Corollary 9.2. □

Theorem 9.8 *Suppose that c is real and positive, and that the integers m and n are non-negative. Let*

$$\Psi_n(y) = \int_0^y \psi_n(x)\;dx\,. \tag{204}$$

If n is odd and m is even, then

$$\int_{-1}^{1}\frac{1}{t}\,\psi_n(t)\,\psi_m(t)\;dt \tag{205}$$

$$= \;i\,c\,\frac{2\,\lambda_m\,\lambda_n}{\lambda_n^2+\lambda_m^2}\,\Psi_n(1)\,\Psi_m(1) \tag{206}$$

$$+\,2\,\frac{\lambda_m}{\lambda_n^2+\lambda_m^2}\,\Psi_m(1)\int_{-1}^{1}\frac{1}{t}\,\psi_n(t)\;dt. \tag{207}$$

If $m = n \pmod 2$, *then*

$$\int_{-1}^{1} \frac{1}{t}\, \psi_n(t)\, \psi_m(t)\ dt = 0\,. \tag{208}$$

Proof. We start with the identity

$$\lambda_n\, \psi_n(x) = \int_{-1}^{1} e^{icxt}\, \psi_n(t)\ dt\,. \tag{209}$$

Integrating (209) with respect to x, we have

$$\begin{aligned}
&\lambda_n \int_0^y \psi_n(x)\, dx \\
&= \int_0^y \left(\int_{-1}^{1} e^{icxt}\, \psi_n(t)\ dt\right)\ dx && (210)\\
&= \int_{-1}^{1} \psi_n(t) \int_0^y e^{ixct}\ dx\, dt && (211)\\
&= \frac{1}{i\,c}\int_{-1}^{1} \frac{1}{t}\, \psi_n(t)\, e^{icyt}\ dt - \frac{1}{i\,c}\int_{-1}^{1} \frac{1}{t}\, \psi_n(t)\ dt\,, && (212)
\end{aligned}$$

which we summarize as

$$\lambda_n\, \Psi_n(y) = \frac{1}{i\,c}\int_{-1}^{1} \frac{1}{t}\, \psi_n(t)\, e^{icyt}\ dt - \frac{1}{i\,c}\int_{-1}^{1} \frac{1}{t}\, \psi_n(t)\ dt\,. \tag{213}$$

Taking the inner product of (213) and $\psi_m(y)$, we obtain

$$\begin{aligned}
&\lambda_n \int_{-1}^{1} \Psi_n(y)\, \psi_m(y)\ dy \\
&= \frac{1}{i\,c}\int_{-1}^{1} \psi_m(y)\cdot\left(\int_{-1}^{1} \frac{1}{t}\, \psi_n(t)\, e^{icyt}\ dt\right)\ dy \\
&\quad -\frac{1}{i\,c}\int_{-1}^{1} \psi_m(y)\cdot\left(\int_{-1}^{1} \frac{1}{t}\, \psi_n(t)\ dt\right)\ dy && (214)\\
&= \frac{1}{i\,c}\int_{-1}^{1} \frac{1}{t}\, \psi_n(t)\cdot\left(\int_{-1}^{1} e^{icyt}\, \psi_m(y)\ dy\right)\ dt \\
&\quad -\frac{1}{i\,c}\int_{-1}^{1} \frac{1}{t}\, \psi_n(t)\ dt\cdot\int_{-1}^{1} \psi_m(y)\ dy && (215)\\
&= \frac{\lambda_m}{i\,c}\int_{-1}^{1} \frac{1}{t}\, \psi_n(t)\, \psi_m(t)\ dt \\
&\quad -\frac{1}{i\,c}\int_{-1}^{1} \frac{1}{t}\, \psi_n(t)\ dt\cdot\int_{-1}^{1} \psi_m(y)\ dy\,, && (216)
\end{aligned}$$

which we summarize as

$$\begin{aligned}
&\int_{-1}^{1} \frac{1}{t}\, \psi_n(t)\, \psi_m(t)\; dt \\
&\quad = \; i\, c\, \frac{\lambda_n}{\lambda_m} \int_{-1}^{1} \Psi_n(t)\, \psi_m(t)\; dt \\
&\qquad + \frac{1}{\lambda_m} \int_{-1}^{1} \frac{1}{t}\, \psi_n(t)\; dt \cdot \int_{-1}^{1} \psi_m(y)\; dy
\end{aligned} \tag{217}$$

Exchanging m with n, we convert (217) into

$$\begin{aligned}
&\int_{-1}^{1} \frac{1}{t}\, \psi_m(t)\, \psi_n(t)\; dt \\
&\quad = \; i\, c\, \frac{\lambda_m}{\lambda_n} \int_{-1}^{1} \Psi_m(t)\, \psi_n(t)\; dt \\
&\qquad + \frac{1}{\lambda_n} \int_{-1}^{1} \frac{1}{t}\, \psi_m(t)\; dt \cdot \int_{-1}^{1} \psi_n(y)\; dy\,,
\end{aligned} \tag{218}$$

and combining (217), (218), we get

$$\begin{aligned}
&\frac{\lambda_n}{\lambda_m}\, i\, c \int_{-1}^{1} \Psi_n(t)\, \psi_m(t)\; dt - \frac{\lambda_m}{\lambda_n}\, i\, c \int_{-1}^{1} \Psi_m(t)\, \psi_n(t)\; dt \\
&\quad = \; \frac{1}{\lambda_n} \int_{-1}^{1} \frac{1}{t}\, \psi_m(t)\; dt \cdot \int_{-1}^{1} \psi_n(t)\; dt \\
&\qquad - \frac{1}{\lambda_m} \int_{-1}^{1} \frac{1}{t}\, \psi_n(t)\; dt \cdot \int_{-1}^{1} \psi_m(t)\; dt\,.
\end{aligned} \tag{219}$$

Suppose that m is even and n is odd; then the first product in the right hand side of (219) is zero, so

$$\begin{aligned}
&\frac{\lambda_n}{\lambda_m}\, i\, c \int_{-1}^{1} \Psi_n(t)\, \psi_m(t)\; dt - \frac{\lambda_m}{\lambda_n}\, i\, c \int_{-1}^{1} \Psi_m(t)\, \psi_n(t)\; dt \\
&\quad = \; - \frac{1}{\lambda_m} \int_{-1}^{1} \frac{1}{t}\, \psi_n(t)\; dt \cdot \int_{-1}^{1} \psi_m(t)\; dt\,,
\end{aligned} \tag{220}$$

which is equivalent to

$$\begin{aligned}
&\int_{-1}^{1} \Psi_n(t)\, \psi_m(t)\; dt \\
&\quad = \; \frac{\lambda_m^2}{\lambda_n^2} \int_{-1}^{1} \Psi_m(t)\, \psi_n(t)\; dt \\
&\qquad - \frac{1}{\lambda_n}\, \frac{1}{i\, c} \int_{-1}^{1} \frac{1}{t}\, \psi_n(t)\; dt \cdot \int_{-1}^{1} \psi_m(t)\; dt\,,
\end{aligned} \tag{221}$$

or

$$\begin{aligned}
&\int_{-1}^{1} \Psi_m(t)\,\psi_n(t)\;dt \\
&\quad= \frac{\lambda_n^2}{\lambda_m^2}\int_{-1}^{1} \Psi_n(t)\,\psi_m(t)\;dt \\
&\qquad+\frac{\lambda_n}{\lambda_m^2}\,\frac{1}{i\,c}\int_{-1}^{1} \frac{1}{t}\,\psi_n(t)\;dt\cdot\int_{-1}^{1}\psi_m(t)\;dt\,.
\end{aligned} \tag{222}$$

On the other hand, integrating the left side of (222) by parts, we obtain

$$\begin{aligned}
&\int_{-1}^{1} \Psi_m(t)\,\psi_n(t)\;dt \\
&\quad= \Psi_n(1)\,\Psi_m(1)-\Psi_n(-1)\,\Psi_m(-1)-\int_{-1}^{1}\Psi_n(t)\,\psi_m(t)\;dt\,.
\end{aligned} \tag{223}$$

Since the product $\Psi_m(x)\,\Psi_n(x)$ is an odd function when $m \neq n \pmod 2$, we rewrite (223) as

$$\begin{aligned}
&\int_{-1}^{1} \Psi_m(t)\,\psi_n(t)\;dt \\
&\quad= 2\,\Psi_n(1)\,\Psi_m(1)-\int_{-1}^{1}\Psi_n(t)\,\psi_m(t)\;dt\,.
\end{aligned} \tag{224}$$

The combination of (222) and (224) implies that

$$\begin{aligned}
&\int_{-1}^{1} \Psi_n(t)\,\psi_m(t)\;dt+\frac{\lambda_n^2}{\lambda_m^2}\int_{-1}^{1}\Psi_n(t)\,\psi_m(t)\;dt \\
&\quad= 2\,\Psi_n(1)\,\Psi_m(1)-\frac{\lambda_n}{\lambda_m^2}\,\frac{1}{i\,c}\int_{-1}^{1}\frac{1}{t}\,\psi_n(t)\;dt\cdot\int_{-1}^{1}\psi_m(t)\;dt\,,
\end{aligned} \tag{225}$$

or

$$\begin{aligned}
&\frac{\lambda_m^2+\lambda_n^2}{\lambda_m^2}\int_{-1}^{1} \Psi_n(t)\,\psi_m(t)\;dt \\
&\quad= 2\,\Psi_n(1)\,\Psi_m(1)-\frac{\lambda_n}{\lambda_m^2}\,\frac{1}{i\,c}\int_{-1}^{1}\frac{1}{t}\,\psi_n(t)\;dt\cdot\int_{-1}^{1}\psi_m(t)\;dt\,,
\end{aligned} \tag{226}$$

which is equivalent to

$$\begin{aligned}
&\int_{-1}^{1} \Psi_n(t)\,\psi_m(t)\;dt \\
&\quad= \frac{2\,\lambda_m^2}{\lambda_n^2+\lambda_m^2}\,\Psi_n(1)\,\Psi_m(1) \\
&\qquad-\frac{\lambda_n}{\lambda_n^2+\lambda_m^2}\,\frac{1}{i\,c}\int_{-1}^{1}\frac{1}{t}\,\psi_n(t)\;dt\cdot\int_{-1}^{1}\psi_m(t)\;dt\,.
\end{aligned} \tag{227}$$

Finally, combining (217) and (227), we have

$$\begin{aligned}\int_{-1}^{1} \frac{1}{t}\,\psi_n(t)\,\psi_m(t)\;dt & \\ = \; & i\,c\,\frac{2\,\lambda_m\,\lambda_n}{\lambda_n^2+\lambda_m^2}\,\Psi_n(1)\,\Psi_m(1) \\ & + \frac{\lambda_m}{\lambda_n^2+\lambda_m^2}\int_{-1}^{1}\frac{1}{t}\,\psi_n(t)\;dt\cdot\int_{-1}^{1}\psi_m(t)\;dt\,. \qquad (228)\end{aligned}$$

Equation (208) is easily proven since the product $\frac{1}{t}\,\psi_m(x)\,\psi_n(x)$ is an odd function whenever $m = n \pmod 2$. □

The above theorems do not use much of the detailed structure of the integral operators of which the functions $\{\psi_j\}$ are eigenfunctions. Thus many of them generalize easily to the case of an operator $L : L^2[0,1] \to L^2[0,1]$ defined via the formula

$$L(\psi)(x) = \int_0^1 K(xt)\,\psi(t)\;dt, \qquad (229)$$

for some function $K : [0,1] \to \mathbb{C}$; the following theorem is an example of this.

Theorem 9.9 *Let λ_1, λ_2 be two eigenvalues of the operator L defined by (229), that is,*

$$\int_0^1 K(xt)\,\psi_1(t)\;dt \;=\; \lambda_1\psi_1(x), \qquad (230)$$

$$\int_0^1 K(xt)\,\psi_2(t)\;dt \;=\; \lambda_2\psi_2(x). \qquad (231)$$

Then

$$\frac{\lambda_2}{\lambda_1} = \frac{\displaystyle\int_0^1 x\,\psi_1'(x)\,\psi_2(x)\;dx}{\displaystyle\int_0^1 x\,\psi_2'(x)\,\psi_1(x)\;dx}, \qquad (232)$$

provided that neither λ_1 nor the denominator of the right hand side of (232) is zero.

Proof. Differentiating (230), (231) with respect to x, we get

$$\int_0^1 t\,K'(xt)\,\psi_1(t)\;dt \;=\; \lambda_1\psi_1'(x), \qquad (233)$$

$$\int_0^1 t\,K'(xt)\,\psi_2(t)\;dt \;=\; \lambda_2\psi_2'(x). \qquad (234)$$

Multiplying (233) by $x\,\psi_2(x)$, we have

$$\lambda_1\,x\,\psi_1'(x)\,\psi_2(x) = x\,\psi_2(x)\int_0^1 t\,K'(xt)\,\psi_1(t)\;dt. \qquad (235)$$

Integrating on the interval $[0,1]$, we obtain

$$\lambda_1 \int_0^1 x\,\psi_1'(x)\,\psi_2(x)\;dx = \int_0^1 x\,\psi_2(x) \int_0^1 t\,K'(xt)\,\psi_1(t)\;dt\,dx \tag{236}$$

$$= \int_0^1 t\,\psi_1(t) \int_0^1 x\,K'(xt)\,\psi_2(x)\;dx\,dt. \tag{237}$$

Renaming the variables of integration on the right hand side from x to t and vice versa, we get

$$\lambda_1 \int_0^1 x\,\psi_1'(x)\,\psi_2(x)\;dx = \int_0^1 x\,\psi_1(x) \int_0^1 t\,K'(xt)\,\psi_2(t)\;dt\,dx. \tag{238}$$

Substituting (234) into (238), we obtain

$$\lambda_1 \int_0^1 x\,\psi_1'(x)\,\psi_2(x)\;dx = \lambda_2 \int_0^1 x\,\psi_1(x)\,\psi_2'(x)\,dx, \tag{239}$$

from which (232) follows immediately, as does its caveat. □

The following theorem establishes the relation between the norm of each function ψ_j on $[-1,1]$ (which in this paper is taken to be one), and its norm on $(-\infty,\infty)$.

Theorem 9.10 *Suppose that c is real and positive, and that the integer n is non-negative. Then*

$$\int_{-\infty}^{\infty} \psi_n^2(x)\;dx = \frac{1}{\mu_n}. \tag{240}$$

where μ_n is given by (21).

Proof.

$$\begin{aligned}\int_{-\infty}^{\infty} \psi_n^2(x)\;dx &= \int_{-\infty}^{\infty} \left(\frac{1}{\pi\mu_n}\int_{-1}^{1} \psi_n(t)\,\frac{\sin(c\cdot(x-t))}{x-t}\;dt\right)\psi_n(x)\;dx \\ &= \frac{1}{\mu_n}\int_{-1}^{1} \psi_n(t)\cdot\left(\frac{1}{\pi}\int_{-\infty}^{\infty} \frac{\sin(c\cdot(x-t))}{x-t}\,\psi_n(x)\;dx\right)dt \\ &= \frac{1}{\mu_n}\int_{-1}^{1} \psi_n^2(t)\;dt \\ &= \frac{1}{\mu_n}.\end{aligned}$$

□

The following theorem extends Theorem (9.10) to any band-limited function with band limit c.

Theorem 9.11 *Suppose that c is real and positive, that the integer n is non-negative, and that $f : \mathbb{R} \to \mathbb{C}$ is a band-limited function with band limit c. Then*

$$\int_{-\infty}^{\infty} \psi_n(x)\, f(x)\; dx = \frac{1}{\mu_n} \int_{-1}^{1} \psi_n(x)\, f(x)\; dx. \tag{241}$$

Proof.

$$\begin{aligned} &\int_{-\infty}^{\infty} \psi_n(x)\, f(x)\; dx \\ = &\int_{-\infty}^{\infty} \left(\frac{1}{\pi \mu_n} \int_{-1}^{1} \frac{\sin(c \cdot (x-t)\,)}{x-t}\, \psi_n(t)\; dt \right) f(x)\; dx \\ = &\frac{1}{\mu_n} \int_{-1}^{1} \psi_n(t) \cdot \left(\frac{1}{\pi} \int_{-\infty}^{\infty} \frac{\sin(c \cdot (x-t))}{x-t}\, f(x)\; dx \right) dt \\ = &\frac{1}{\mu_n} \int_{-1}^{1} \psi_n(t)\, f(t)\; dt\,. \end{aligned}$$

□

Theorem 9.12 *Suppose that c is real and positive, and that the integer n is non-negative. Then*

$$\int_{-\infty}^{\infty} e^{icxt} \psi_m(t) dt = \begin{cases} \dfrac{\lambda_m}{\mu_m} \psi_m(x), & \text{if } -1 < x < 1, \\ 0, & \text{if } x > 1 \text{ or } x < -1. \end{cases} \tag{242}$$

Proof. Since ψ_m is an eigenfunction of the operator Q_c defined in (19), and μ_m is the corresponding eigenvalue,

$$\mu_m \psi_m(t) = \frac{1}{\pi} \int_{-1}^{1} \frac{sin(c \cdot (x-u))}{x-u}\, \psi_m(u)\; du. \tag{243}$$

Thus

$$\int_{-\infty}^{\infty} e^{icxt}\, \psi_m(t)\; dt$$

$$= \frac{1}{\mu_m} \int_{-\infty}^{\infty} e^{icxt} \left(\frac{1}{\pi} \int_{-1}^{1} \frac{sin(c \cdot (x-u))}{x-u}\, \psi_m(u)\; du \right)\; dt \tag{244}$$

$$= \frac{1}{\mu_m} \int_{-1}^{1} \psi_m(u) \left(\frac{1}{\pi} \int_{-\infty}^{\infty} \frac{sin(c \cdot (x-u))}{x-u}\, e^{icxt}\; dt \right)\; du \tag{245}$$

Since the innermost integral is the orthogonal projection operator onto the space of functions of band limit c on $(-\infty,\infty)$, applied to the function e^{icxt}, it follows that:

$$\begin{aligned}
&\int_{-\infty}^{\infty} e^{icxt}\,\psi_m(t)\;dt \\
&= \frac{1}{\mu_m}\int_{-1}^{1}\psi_m(u)\left(\left\{\begin{array}{ll} e^{icxu}, & \text{if } -1<x<1, \\ 0, & \text{if } x>1 \text{ or } x<-1 \end{array}\right\}\right) du \qquad (246) \\
&= \left\{\begin{array}{ll} \dfrac{1}{\mu_m}\displaystyle\int_{-1}^{1}\psi_m(u)\,e^{icxu}\,du, & \text{if } -1<x<1, \\ 0, & \text{if } x>1 \text{ or } x<-1, \end{array}\right. \qquad (247)
\end{aligned}$$

from which (242) follows immediately. □

The following five theorems establish formulae for the derivatives of prolate functions and their associated eigenvalues with respect to c.

Theorem 9.13 *For all positive real c and non-negative integer m,*

$$\frac{\partial\lambda_m}{\partial c} = \lambda_m\,\frac{2\,\psi_m^2(1)-1}{2\,c}. \qquad (248)$$

Proof. We start with

$$\lambda_m\psi_m(x) = \int_{-1}^{1} e^{icxt}\,\psi_m(t)\;dt. \qquad (249)$$

Differentiating (249) with respect to c, we obtain

$$\begin{aligned}
&\frac{\partial\lambda_m}{\partial c}\,\psi_m(x) + \lambda_m\,\frac{\partial\psi_m(x)}{\partial c} \\
&= \int_{-1}^{1} i\,x\,t\,e^{icxt}\,\psi_m(t)\;dt + \int_{-1}^{1} e^{icxt}\,\frac{\partial\psi_m(t)}{\partial c}\;dt. \qquad (250)
\end{aligned}$$

Multiplying by $\psi_m(x)$ on both sides of (250), and integrating on the interval $[-1,1]$, we get

$$\begin{aligned}
&\int_{-1}^{1}\psi_m(x)\left(\frac{\partial\lambda_m}{\partial c}\,\psi_m(x) + \lambda_m\,\frac{\partial\psi_m(x)}{\partial c}\right)dx \qquad (251) \\
&= \int_{-1}^{1}\psi_m(x)\int_{-1}^{1} i\,x\,t\,e^{icxt}\,\psi_m(t)\;dt\;dx \\
&\quad + \int_{-1}^{1}\psi_m(x)\int_{-1}^{1} e^{icxt}\,\frac{\partial\psi_m(t)}{\partial c}\;dt\;dx,
\end{aligned}$$

which we rewrite as

$$\begin{aligned}
&\frac{\partial\lambda_m}{\partial c} + \lambda_m \int_{-1}^{1} \frac{\partial\psi_m(x)}{\partial c}\, \psi_m(x)\; dx \\
&\quad= \int_{-1}^{1} i\, t\, \psi_m(t) \int_{-1}^{1} e^{icxt}\, x\, \psi_m(x)\; dx\; dt \\
&\qquad+ \int_{-1}^{1} \frac{\partial\psi_m(t)}{\partial c} \int_{-1}^{1} e^{icxt}\, \psi_m(x)\; dx\; dt && (252) \\
&\quad= \lambda_m \int_{-1}^{1} i\, t\, \psi_m(t)\, \frac{1}{i\, c}\, \frac{\partial\psi_m(t)}{\partial t}\; dt \\
&\qquad+ \lambda_m \int_{-1}^{1} \frac{\partial\psi_m(t)}{\partial c}\, \psi_m(t)\; dt\,, && (253)
\end{aligned}$$

which we summarize as

$$\frac{\partial\lambda_m}{\partial c} = \frac{\lambda_m}{c} \int_{-1}^{1} t\, \psi_m(t)\, \frac{\partial\psi_m(t)}{\partial t}\; dt\,. \tag{254}$$

On the other hand, integrating the right-hand side of (254) by parts, we have

$$\begin{aligned}
&\int_{-1}^{1} t\, \psi_m(t)\, \frac{\partial\psi_m(t)}{\partial t}\; dt && (255) \\
&\quad= \psi_m^2(1) + \psi_m^2(-1) - 1 - \int_{-1}^{1} \psi_m(t)\, t\, \frac{\partial\psi_m(t)}{\partial t}\; dt\,,
\end{aligned}$$

which we rewrite as

$$\int_{-1}^{1} t\, \psi_m(t)\, \frac{\partial\psi_m(t)}{\partial t}\; dt = \psi_m^2(1) - \frac{1}{2}\,. \tag{256}$$

Finally, substituting (256) into (254), we get

$$\frac{\partial\lambda_m}{\partial c} = \lambda_m\, \frac{2\, \psi_m^2(1) - 1}{2\, c}\,. \tag{257}$$

□

Theorem 9.14 *For any positive real c and non-negative integer m,*

$$\frac{\partial\mu_m}{\partial c} = \frac{2}{c}\, \mu_m\, \psi_m^2(1)\,. \tag{258}$$

Proof. We start with the identity

$$\mu_m = \frac{2c}{\pi}\,\overline{\lambda}_m\,\lambda_m. \tag{259}$$

Differentiating (259) with respect to c, we get

$$\frac{\partial \mu_m}{\partial c} = \frac{2c}{\pi}\left(\overline{\lambda}_m\,\frac{\partial \lambda_m}{\partial c} + \lambda_m\,\frac{\partial \overline{\lambda}_m}{\partial c}\right) + \frac{2}{\pi}\,\overline{\lambda}_m\,\lambda_m\,. \tag{260}$$

Substituting Lemma 9.13 into (260), we get

$$\begin{aligned}
\frac{\partial \mu_m}{\partial c} &= \frac{2c}{\pi}\cdot 2\,\overline{\lambda}_m\,\lambda_m\,\frac{2\,\psi_m^2(1)-1}{2\,c} + \frac{2}{\pi}\,\overline{\lambda}_m\,\lambda_m \qquad (261)\\
&= 2\,\mu_m\,\frac{2\,\psi_m^2(1)-1}{2\,c} + \frac{1}{c}\,\mu_m\\
&= \frac{2}{c}\,\mu_m\,\psi_m^2(1) - \frac{1}{c}\,\mu_m + \frac{1}{c}\,\mu_m\\
&= \frac{2}{c}\,\mu_m\,\psi_m^2(1)\,. \qquad (262)
\end{aligned}$$

□

The following theorem immediately follows from Theorems 9.13 and 9.14.

Theorem 9.15 *For all positive real c and non-negative integer m, n,*

$$\left(\frac{\lambda_m}{\lambda_n}\right)' = \frac{\lambda_m}{\lambda_n}\,\frac{1}{c}\,\left(\psi_m^2(1) - \psi_n^2(1)\right)\,, \tag{263}$$

$$\left(\frac{\mu_m}{\mu_n}\right)' = \frac{\mu_m}{\mu_n}\,\frac{2}{c}\,\left(\psi_m^2(1) - \psi_n^2(1)\right)\,. \tag{264}$$

Theorem 9.16 *Suppose that c is real and positive, and the integers m, n are non-negative. If $m \neq n$, then*

$$\int_{-1}^{1} \psi_m(t)\,\frac{\partial \psi_n}{\partial c}(t)\;dt = -\frac{2}{c}\,\frac{\lambda_n\,\lambda_m}{\lambda_m^2 - \lambda_n^2}\,\psi_m(1)\,\psi_n(1)\,. \tag{265}$$

If $m = n$, then

$$\int_{-1}^{1} \psi_m(t)\,\frac{\partial \psi_n}{\partial c}(t)\;dt = 0. \tag{266}$$

Proof. Since the norm of ψ_n on $[-1,1]$ remains constant as c varies, ψ_n must be orthogonal on $[-1,1]$ to its own derivative with respect to c, which immediately yields (266). To establish (265), we start with the identity

$$\lambda_n \psi_n(x) = \int_{-1}^{1} e^{icxt}\, \psi_n(t)\, dt\,. \tag{267}$$

Differentiating (267) with respect to c, we get

$$\begin{aligned} &\frac{\partial \lambda_n}{\partial c}\, \psi_n(x) + \lambda_n \frac{\partial \psi_n}{\partial c} \\ &= \int_{-1}^{1} \left(i\, x\, t\, e^{icxt}\, \psi_n(t) + e^{icxt}\, \frac{\partial \psi_n(t)}{\partial c} \right)\, dt\,. \end{aligned} \tag{268}$$

Multiplying both sides of (268) by $\psi_m(x)$ and integrating with respect to x, we have

$$\begin{aligned} &\lambda_n \int_{-1}^{1} \psi_m(x)\, \frac{\partial \psi_n(x)}{\partial c}\, dx \\ &= \frac{\lambda_n}{c} \int_{-1}^{1} x\, \psi_n'(x)\, \psi_m(x)\, dx + \lambda_m \int_{-1}^{1} \psi_m(t)\, \frac{\partial \psi_n(t)}{\partial c}\, dt\,, \end{aligned} \tag{269}$$

which, using (176), we rewrite as

$$\begin{aligned} &(\lambda_n - \lambda_m) \int_{-1}^{1} \psi_m(t)\, \frac{\partial \psi_n(t)}{\partial c}\, dt \\ &= \frac{\lambda_n}{c}\, \frac{\lambda_m}{\lambda_m + \lambda_n}\, (2\, \psi_m(1)\, \psi_n(1) - \delta_{mn})\,. \end{aligned} \tag{270}$$

Assuming that $m \neq n$, and dividing by $\lambda_n - \lambda_m$, we then get (265). □

Theorem 9.17 *Suppose that c is real and positive, and the integer m is non-negative. Then*

$$\frac{\partial \chi_m}{\partial c} = 2c \int_{-1}^{1} x^2 \psi_m^2(x). \tag{271}$$

Proof. Due to Theorem 2.6,

$$(1-x^2)\psi_m''(x) - 2x\psi_m'(x) + (\chi_m - c^2x^2)\, \psi_m(x) = 0. \tag{272}$$

Making the infinitesimal changes $c = c + h$, $\chi_m = \chi_m + \varepsilon$, and $\psi_m(x) = \psi_m(x) + \delta(x)$, this becomes

$$\begin{aligned} &(1-x^2) \cdot (\psi_m''(x) + \delta''(x)) - 2x \cdot (\psi_m'(x) + \delta'(x)) \\ &\quad + (\chi_m + \varepsilon - (c+h)^2x^2) \cdot (\psi_m(x) + \delta(x)) = 0. \end{aligned} \tag{273}$$

Expanding each term, discarding infinitesimals of the second order or greater (that is, products of two or more of the quantities h, ε, and $\delta(x)$), and subtracting (272), we get

$$(1-x^2)\,\delta''(x) - 2x\delta'(x) + (\chi_m - c^2x^2)\,\delta(x) + (\varepsilon - 2chx^2)\psi_m(x) = 0. \tag{274}$$

Let the self-adjoint differential operator L be defined by the formula

$$L(f)(x) = (1-x^2)f''(x) - 2xf'(x) + (\chi_m - c^2x^2)f(x). \tag{275}$$

Then, multiplying (274) by $\psi_m(x)/h$ and integrating on $[-1,1]$, we get

$$\int_{-1}^{1} L(\frac{\partial\psi_m}{\partial c})(x)\,\psi_m(x)\;dx + \frac{\varepsilon}{h} - \int_{-1}^{1} 2cx^2\psi_m^2(x) = 0. \tag{276}$$

Now $\frac{\varepsilon}{h} = \frac{\partial\chi_m}{\partial c}$. In addition, since L is self-adjoint,

$$\int_{-1}^{1} L(\frac{\partial\psi_m}{\partial c})(x)\,\psi_m(x)\;dx = \int_{-1}^{1} \frac{\partial\psi_m}{\partial c}(x)\,L(\psi_m)(x)\;dx. \tag{277}$$

But due to (272), $L(\psi_m)(x) = 0$ for all $x \in [-1,1]$, so the integral (277) is zero. Thus (276) becomes

$$\frac{\partial\chi_m}{\partial c} = 2c\int_{-1}^{1} x^2\psi_m^2(x). \tag{278}$$

□

10 Generalizations and Conclusions

In this paper, we design quadrature rules for band-limited functions, based on the properties of Prolate Spheroidal Wave Functions (PSWFs), and the connections of the latter with certain fundamental integral operators (see (17), (19) in Section 2.5). The quadratures are a surprisingly close analogue for band-limited functions of Gaussian quadratures for polynomials, in that they have positive weights, are optimal in the appropriately defined sense, and their nodes, when used for approximation (as opposed to integration), result in extremely efficient interpolation formulae. Thus, Sections 5-7 of this paper can be viewed as reproducing for band-limited functions much of the standard polynomial-based approximation theory (for which see, for example, [24]). Generally, there is a striking analogy between the band-limited functions and polynomials.

Obviously, there are certain differences between the resulting apparatus and the standard numerical analysis. To start with, where the classical techniques are optimal for polynomials, the approach of this paper is optimal for band-limited functions; whenever the functions to be dealt with are naturally represented by trigonometric expansions on finite intervals, our quadrature and interpolation formulae tend to be more efficient than those based on the polynomials. When the functions to be dealt with are naturally represented by polynomials, the classical approach is more efficient; however, many physical phenomena involve bend-limited functions, and very few involve polynomials.

Qualitatively, the quadrature (and interpolation) nodes obtained in this paper behave like a compromise between the Gaussian nodes and the equispaced ones: near the middle of the interval, they are very nearly equispaced, and near the ends, they concentrate somewhat, but much less than the Gaussian (or Chebychev) nodes do. For large c, the distance between nodes near the ends of the interval is of the order $\frac{1}{c^{3/2}}$, with the total number of nodes close to $\frac{c}{\pi}$. In contrast, the distance between the Gaussian nodes near the ends of the interval is of the order $\frac{1}{n^2}$, with n the total number of nodes. A closely related phenomenon is the reduced norm of the differentiation operator based on the prolate expansions: for an $n-$point differentiation formula, the norm is of the order $n^{3/2}$, as opposed to n^2 for polynomial-based spectral differentiation. Thus, PSWFs are likely to be a better tool for the design of spectral and pseudo-spectral techniques than the orthogonal polynomials and related functions.

Much of the analytical apparatus we use was developed more than 30 years ago (see [20]-[21], [17], [18]); the fundamental importance of these results in certain areas of electrical engineering and physics has also been understood for a long time. However, there appears to have been no prior attempt made to view band-limited functions as a source of *numerical* algorithms. Generally, there is a fairly limited amount of information in the literature about the PSWFs, especially when compared to the wealth of facts on many other special functions. Section 9 of this paper is an attempt to remedy this situation to a small degree.

The apparatus built in this paper is a strictly one-dimensional one. Obviously, one can construct discretizations of rectangles, cubes, etc. by using direct products of one-dimensional grids; the resulting numerical algorithms are satisfactory but not optimal. Furthermore, representation of band-limited functions on regions in higher dimensions is of both theoretical and engineering interest. Obvious applications include seismic data collection and processing, antenna theory, NMR imaging, and many others. When the region of interest is a sphere, most of the necessary analytical apparatus can be found in [21]. At the present time, we have constructed and implemented somewhat rudimentary versions of the relevant numerical algorithms; we are conducting numerical experiments with these, and will report the results at a later date. A much more difficult set of questions is presented by the structure of band-limited functions on more general regions.

References

[1] C.J. Bouwkamp, *On Spheroidal Wave Functions of Order Zero*, J. Math. Phys., 26, 1947, pp. 79-92.

[2] H. Cheng, N. Yarvin, V. Rokhlin, *Non-Linear Optimization, Quadrature, and Interpolation*, Yale University Technical Report, YALEU/DCS/RR-1169, 1998, to appear in the SIAM Journal of Non-linear Optimization.

[3] L. Debnath, *Integral Transforms and Their Applications*, CRC Press, New York, 1995.

[4] G. Gripenberg, S.O. Londen, O. Staffans, *Volterra Integral and Functional Equations, Encyclopedia of Mathematics and its Applications*, Cambridge University Press, 1990.

[5] C. Flammer, *Spheroidal Wave Functions*, Stanford University Press, Stanford, Ca, 1956.

[6] F. GANTMACHER AND M. KREIN, *Oscillation matrices and kernels and small oscillations of mechanical systems*, 2nd ed., Gosudarstv. Izdat. Tehn-Teor. Lit., Moscow, 1950 (Russian).

[7] F. A. Grünbaum, *Toeplitz Matrices Commuting With Tridiagonal Matrices*, J. Linear Alg. and Appl., 40, (1981).

[8] F. A. Grünbaum, *Eigenvectors of a Toeplitz Matrix: Discrete Version of the Prolate Spheroidal Wave Functions*, SIAM J. Alg. Disc. Meth., 2(1981).

[9] F. A. Grünbaum, L. Longhi, M. Perlstadt, *Differential Operators Commuting with Finite Convolution Integral Operators: Some Non-Abelian Examples*, SIAM J. Appl. Math. 42(1982).

[10] S. KARLIN, *The Existence of Eigenvalues for Integral Operators*, Trans. Am. Math. Soc. v. 113, pp. 1-17 (1964).

[11] S. KARLIN, AND W. J. STUDDEN, *Tchebycheff Systems with Applications In Analysis And Statistics*, John Wiley (Interscience), New York, 1966.

[12] M. G. KREIN, *The Ideas of P. L. Chebyshev and A. A. Markov in the Theory Of Limiting Values Of Integrals*, American Mathematical Society Translations, Ser. 2, Vol. 12, 1959, pp. 1-122.

[13] J. MA, V. ROKHLIN, AND S. WANDZURA, *Generalized Gaussian Quadratures For Systems of Arbitrary Functions*, SIAM Journal of Numerical Analysis, v. 33, No. 3, pp. 971-996, 1996.

[14] A. A. MARKOV, *On the limiting values of integrals in connection with interpolation*, Zap. Imp. Akad. Nauk. Fiz.-Mat. Otd. (8) 6 (1898), no.5 (Russian), pp. 146-230 of [15].

[15] A. A. MARKOV, *Selected papers on continued fractions and the theory of functions deviating least from zero*, OGIZ, Moscow-Leningrad, 1948 (Russian).

[16] P.M. Morse. H. Feshbach, *Methods of Theoretical Physics*, McGraw-Hill, New York, 1953.

[17] H.J. Landau. H.O. Pollak, *Prolate Spheroidal Wave Functions, Fourier Analysis, and Uncertainty - II*, The Bell System Technical Journal, January 1961.

[18] H.J. Landau. H.O. Pollak, *Prolate Spheroidal Wave Functions, Fourier Analysis, and Uncertainty - III: The Dimension of Space of Essentially Time- and Band-Limited Signals*. The Bell System Technical Journal, July 1962.

[19] H.J. Landau. H. Widom, *Eigenvalue Distribution of Time and Frequency Limiting*, Journal of Mathematical Analysis and Applications, 77, 469-481 (1980).

[20] D. Slepian. H.O. Pollak, *Prolate Spheroidal Wave Functions, Fourier Analysis, and Uncertainty - I*. The Bell System Technical Journal, January 1961.

[21] D. Slepian. *Prolate Spheroidal Wave Functions, Fourier Analysis, and Uncertainty - IV: Extensions to Many Dimensions, Generalized Prolate Spheroidal Wave Functions*, The Bell System Technical Journal, November 1964.

[22] D. Slepian. *Prolate Spheroidal Wave Functions, Fourier Analysis, and Uncertainty - V: The Discrete Case*, The Bell System Technical Journal, May-June 1978.

[23] D. Slepian, *Some Comments on Fourier Analysis, Uncertainty, and Modeling* SIAM Review, V. 25, No. 3, July 1983.

[24] J. Stoer and R. Bulirsch, *Introduction To Numerical Analysis*, 2'nd ed., Springer-Verlag, 1992

[25] N. Yarvin and V. Rokhlin, *Generalized Gaussian Quadratures and Singular Value Decompositions of Integral Operators*, SIAM Journal of Scientific Computing, Vol. 20, No. 2, pp. 699-718 (1998).

Table 1: Quadrature performance for varying band limits, for $\varepsilon = 10^{-7}$

c	n	Maximum Errors		N_{pol}
		Roots	Refined	
10.0	9	0.96E-05	0.51E-07	13
20.0	13	0.17E-04	0.94E-07	19
30.0	17	0.12E-04	0.50E-07	25
40.0	20	0.70E-05	0.30E-06	31
50.0	24	0.35E-05	0.83E-07	37
60.0	27	0.25E-04	0.27E-06	43
70.0	31	0.11E-04	0.66E-07	48
80.0	34	0.48E-05	0.17E-06	54
90.0	38	0.21E-05	0.40E-07	59
100.0	41	0.12E-04	0.91E-07	65
200.0	74	0.24E-05	0.86E-07	118
300.0	106	0.32E-05	0.21E-06	171
400.0	139	0.52E-05	0.62E-07	223
500.0	171	0.56E-05	0.88E-07	275
600.0	203	0.58E-05	0.11E-06	326
700.0	235	0.57E-05	0.12E-06	377
800.0	267	0.55E-05	0.13E-06	428
900.0	299	0.53E-05	0.14E-06	479
1000.0	331	0.50E-05	0.14E-06	530
1200.0	395	0.44E-05	0.13E-06	632
1400.0	459	0.38E-05	0.11E-06	734
1600.0	523	0.31E-05	0.97E-07	835
1800.0	587	0.28E-05	0.80E-07	937
2000.0	651	0.23E-05	0.64E-07	1038
2400.0	778	0.29E-05	0.15E-06	1240
2800.0	906	0.19E-05	0.84E-07	1442
4000.0	1288	0.37E-05	0.17E-06	2047

Table 2: Quadrature performance for varying precisions, for $c = 50$

ε	n	Maximum Errors		N_{pol}
		Roots	Refined	
0.10E-01	19	0.45E-01	0.10E-01	30
0.10E-02	20	0.70E-02	0.13E-02	32
0.10E-03	21	0.91E-03	0.14E-03	33
0.10E-04	22	0.82E-04	0.13E-04	34
0.10E-05	23	0.54E-04	0.11E-05	36
0.10E-06	24	0.35E-05	0.83E-07	37
0.10E-07	25	0.33E-05	0.57E-08	38
0.10E-08	26	0.18E-06	0.36E-09	39
0.10E-09	26	0.18E-06	0.36E-09	40
0.10E-10	27	0.17E-06	0.21E-10	42
0.10E-11	28	0.79E-08	0.11E-11	43
0.10E-12	29	0.78E-08	0.56E-13	45
0.10E-13	30	0.31E-09	0.27E-14	55

Table 3: Interpolation performance for varying band limits, for $\varepsilon = 10^{-7}$

c	n	Maximum Errors		N_{pol}	
		Roots	Refined	Cheb.	Leg.
5.0	13	0.12E-06	0.12E-06	17	17
10.0	18	0.12E-06	0.13E-06	24	25
15.0	22	0.24E-06	0.25E-06	31	32
20.0	26	0.26E-06	0.28E-06	37	39
25.0	30	0.22E-06	0.23E-06	43	45
30.0	33	0.67E-06	0.73E-06	49	51
35.0	37	0.42E-06	0.46E-06	55	57
40.0	41	0.25E-06	0.27E-06	61	63
45.0	44	0.54E-06	0.60E-06	67	69
50.0	48	0.29E-06	0.33E-06	73	75
100.0	82	0.39E-06	0.46E-06	128	131
150.0	115	0.52E-06	0.64E-06	182	186
200.0	147	0.12E-05	0.15E-05	235	239
250.0	180	0.83E-06	0.11E-05	287	292
300.0	212	0.13E-05	0.17E-05	340	345
350.0	245	0.75E-06	0.10E-05	392	398
400.0	277	0.10E-05	0.14E-05	443	450
450.0	309	0.13E-05	0.18E-05	495	502
500.0	341	0.16E-05	0.22E-05	547	554
1000.0	662	0.16E-05	0.24E-05	1058	1068
1500.0	982	0.15E-05	0.25E-05	1566	1578
2000.0	1301	0.20E-05	0.35E-05	2072	2086

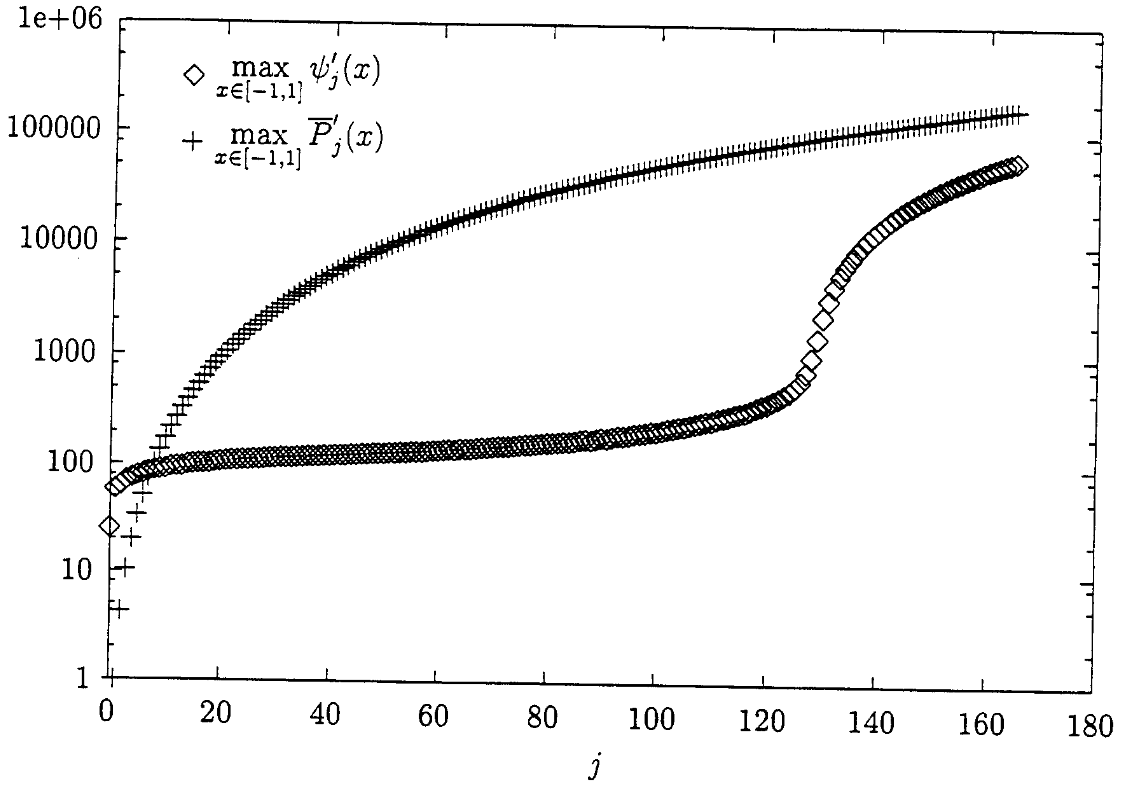
Table 4: Interpolation performance for varying precisions, for $c = 25$

ε	n	Maximum Errors		N_{pol}	
		Roots	Refined	Cheb.	Leg.
0.10E-01	21	0.38E-01	0.43E-01	31	34
0.10E-02	23	0.37E-02	0.41E-02	34	36
0.10E-03	25	0.29E-03	0.31E-03	37	39
0.10E-04	26	0.74E-04	0.81E-04	39	41
0.10E-05	28	0.44E-05	0.47E-05	41	43
0.10E-06	30	0.22E-06	0.23E-06	43	45
0.10E-07	31	0.46E-07	0.49E-07	45	47
0.10E-08	32	0.95E-08	0.10E-07	47	49
0.10E-09	34	0.36E-09	0.38E-09	49	51
0.10E-10	35	0.67E-10	0.70E-10	51	52
0.10E-11	37	0.21E-11	0.22E-11	53	54
0.10E-12	38	0.36E-12	0.37E-12	54	56
0.10E-13	39	0.59E-13	0.63E-13	98	61

Table 5: Quadrature nodes for band-limited functions, with $c = 50$ and $\varepsilon = 10^{-7}$ This table contains only half of the nodes and weights, in particular those for which the node is less than or equal to zero; reflecting these nodes around zero yields the remaining nodes, the weight for the node at $-x$ being the same as the weight for the node at x.

Node	Weight
−.9904522459960804E+00	0.2413064234922188E−01
−.9525601106643832E+00	0.5024347217095568E−01
−.8927960861459153E+00	0.6801787677830858E−01
−.8186117530609125E+00	0.7952155999100788E−01
−.7350624131965875E+00	0.8706680708376023E−01
−.6452878027260844E+00	0.9216240765763570E−01
−.5512554698695428E+00	0.9569254015486106E−01
−.4542505281525226E+00	0.9817257766311556E−01
−.3551568458127944E+00	0.9990914516102242E−01
−.2546173463813596E+00	0.1010880172648715E+00
−.1531287781860989E+00	0.1018214308931439E+00
−.5110121484050418E−01	0.1021735189986602E+00

Figure 2: Maximum norms of derivatives of prolate spheroidal wave functions for $c = 200$, and of normalized Legendre polynomials



Norms of eigenvalues λ_j for $c = 200$:

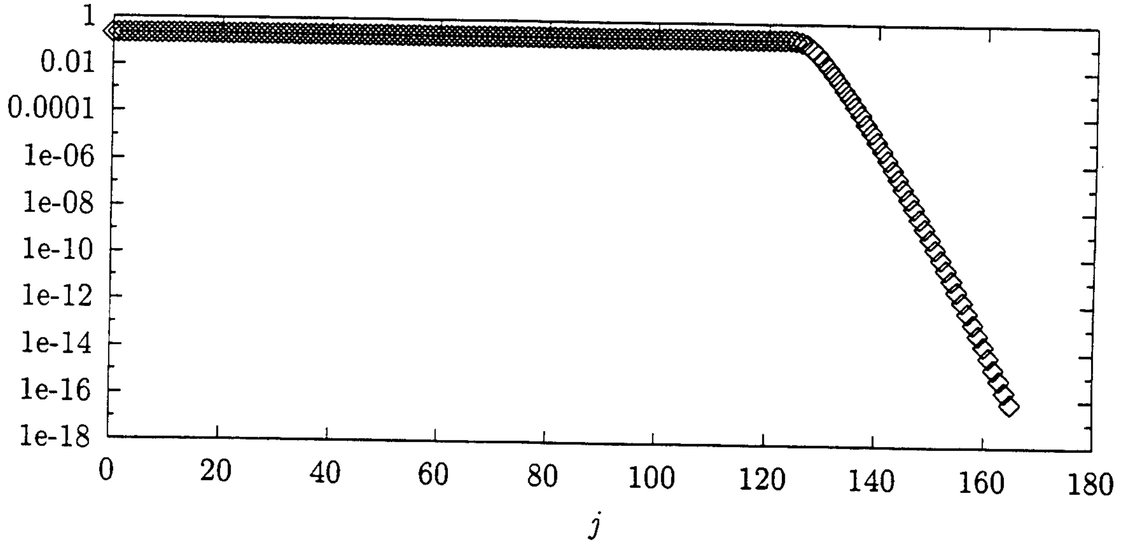


Table 6: Quadrature nodes for band-limited functions, with $c = 150$ and $\varepsilon = 10^{-14}$ This table contains only half of the nodes and weights, in particular those for which the node is less than or equal to zero; reflecting these nodes around zero yields the remaining nodes, the weight for the node at $-x$ being the same as the weight for the node at x.

Node	Weight
−.9982883010959975E+00	0.4374483371752129E−02
−.9911354691596528E+00	0.9842619236149078E−02
−.9788315280982487E+00	0.1463518300250369E−01
−.9621348937901911E+00	0.1862396111287527E−01
−.9418386698454396E+00	0.2184988739217138E−01
−.9186509576802944E+00	0.2442858670932862E−01
−.8931541850293142E+00	0.2648864579258096E−01
−.8658083894041821E+00	0.2814375940413615E−01
−.8369709588254746E+00	0.2948528624795690E−01
−.8069187108185302E+00	0.3058356160435090E−01
−.7758670331396409E+00	0.3149181066633766E−01
−.7439849501152674E+00	0.3225015506203403E−01
−.7114064976175457E+00	0.3288893713079314E−01
−.6782391686910609E+00	0.3343126421620424E−01
−.6445701594098660E+00	0.3389488931551181E−01
−.6104710013384929E+00	0.3429358206877410E−01
−.5760010202980960E+00	0.3463812513892117E−01
−.5412099413257457E+00	0.3493704033879884E−01
−.5061398697742787E+00	0.3519712095895683E−01
−.4708268134473433E+00	0.3542382499917732E−01
−.4353018643598344E+00	0.3562156808557525E−01
−.3995921259242572E+00	0.3579394352776868E−01
−.3637214481257228E+00	0.3594388900778062E−01
−.3277110167114320E+00	0.3607381381247460E−01
−.2915798305819667E+00	0.3618569660385742E−01
−.2553450930388687E+00	0.3628116095737887E−01
−.2190225363501577E+00	0.3636153393399723E−01
−.1826266945721476E+00	0.3642789154364812E−01
−.1461711362450572E+00	0.3648109393796617E−01
−.1096686661347072E+00	0.3652181242257066E−01
−.7313150339365902E−01	0.3655054982303338E−01
−.3657144220122915E−01	0.3656765531685031E−01
0	0.3657333451556860E−01